Port-based System Identification

by

Suthipong Wangpattanasirikul
B.S. Engineering, Cornell University, 1995

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1996

Author .....
Department of Mechanical Engineering
September 1, 1996

Certified by .....
Neville Hogan
Professor
Thesis Supervisor

Accepted by .....
Ain A. Sonin
Chairman, Departmental Committee on Graduate Students

Port-based System Identification

by

Suthipong Wangpattanasirikul

Submitted to the Department of Mechanical Engineering
on September 1, 1996, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

In this thesis, an attempt is made to synthesize a mapping from the observed behaviors of a physical system to its geometric and material parameters. Two different approaches are adopted: one is the standard "black-box" approach; the other is a physical-structure based approach.

A DC torque motor is used as an example system. In this thesis, it is desired to identify four unknown motor parameters: rotor inertia, friction coefficient, torque sensitivity, and rotor winding resistance.

The two identification methods are first used on a set of simulated noise-free data and later verified with an experimental data set. It is found that the conventional linear black-box identification method is inadequate to synthesize a sought-for mapping. Though, generally the approach is powerful in capturing the observed behavior, it does not provide a guide to the choice of the observed input and output which would possibly lead to a unique identification of the physical parameters. So the result from this approach is a non-unique mapping from observed behaviors to motor parameters. Furthermore, the approach could lead to a model order higher than necessary to identify the physical parameters. The excess order in the form of a free integrator could jeopardize the derived model by making it unstable. The physical-structure based approach, on the other hand, provides an appropriate model structure for a particular pair of observed behaviors. The proposed algorithm, based upon the network-structure of the motor, also leads to a set of observed behaviors which can be uniquely mapped into the motor physical parameters. From the simulated data, the estimated parameters are roughly within 5 percent of the actual parameters. When applied to the experimental data, the same approach results in an estimated error no bigger than 12 percent. In addition, the estimated parameters are also used to predict the response of the motor. Overall performance of the prediction is quite satisfactory.

The outcome of the study suggests that, to uniquely map the observed behaviors of a physical system to its geometric and material properties, a knowledge of the physical structure, such as a physical network-structure, is necessary. To effectively extract information about the physical parameters from the observed data, the standard "black-box" approach is inadequate. The proposed network-based algorithm provides a complete map from motor behaviors to motor physical parameters even though it does not fully exploit the property of the network-topology. A further test and recommendation to future work on the algorithm are also given.

Thesis Supervisor: Neville Hogan
Title: Professor

Acknowledgments

Many people have helped this thesis at various points. In particular, I would like to thank Neville, my advisor, for his support, encouragement, and instructions during my stay at MIT. I also would like to thank him for his patient guidance through this thesis. Having an opportunity to work under Neville supervision is one of the great honors and privileges of my academic career.

At the lab, Joe Doeringer, Benjamin Sun, Justin Won, and Michel Lemay were especially helpful to me since day one. I have learned a lot from their suggestions and comments. I owe special thanks to Jeff Chiou, John Madden, and Peter Madden for being there for me during the difficult times.

Neville, Benji, and Peter read the draft of the present work and provided helpful comments. This thesis has profited a lot from them. I also would like to thank all my Thai friends here at MIT. I really enjoyed their company.

Finally, I would like thank my family for their understanding and support. Also, my thanks goes to Prof. Kenneth E Torrance for his insightful comments; I will never forget them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background & Goal

This is an initial study towards a broad goal which is to construct a physically-parameterized model of a physical system from its observed behavior. Specifically, the goal is to synthesize a mapping from physical system behavior to the geometric and material properties used in a model of that behavior. To construct such a map, a natural question to ask is what variables should be measured in mapping from the space of observed behaviors to the corresponding physical parameters. Examining the constitutive equations of a physical system model indicates that if all the variables are measurable, all the parameters should be identifiable. Unfortunately, it is impractical to measure all the behavior variables that have obvious relations to the sought-for parameters. Therefore, each parameter must be estimated entirely from a set of observable behaviors.

The ability to construct a physically-parameterized model of a physical system from its behavior will be useful for practitioners; for example, it would provide a clue to a hardware designer as to what kind of addition and/or modification he or she must make to a particular hardware implementation in order to achieve a certain desired performance.

Given that the technology is getting more sophisticated in response to an increasing demand from consumers, a single piece of hardware may contain several mechanical and electrical components. For instance, a modern professional SLR camera is so complex that it consists of several modules which must be assembled at special plants; these modules-

such as the shutter, finder, and electronics- are then finally assembled together into one camera. Disguised by its small-size working unit, the camera consists of more than 42 million mechanical and electrical combinations. Imagine that the camera is presented to you and you are asked to improve its autofocus operation by 20 percent. What components inside the camera must be modified? The motor in the lens drive unit? Probably but not necessarily. But, to go for the motor idea, what exactly are you looking for in the new motor? How do you find out?

Knowing a physical model of such a camera would certainly help you answer the question. To get such a model is another problem; this is precisely the ultimate goal of this thesis.

As mentioned earlier, it is desired to construct a physically-parameterized model of a physical system from its observed behaviors which in turn are determined by the physical parameters. Therefore, if one can reproduce the features of the system behavior, one takes a step closer to getting the physical parameters. To capture and reproduce the observed behaviors (basically data), in literature and in practice, two lines of approach co-exist: one is the data-driven approach and the other is a physical-principle based approach. Along the data-driven approach, the "black-box" system identification is commonly used. It assumes no prior knowledge of the physical hardware and is based upon the observed system behaviors, the black-box approach attempts to create a model that replicates the observed data. Along the physical-principle based approach, a priori up-to-date knowledge of the hardware is used to construct a model. To show which is better, an interesting example from a book (see [7]) shall be used here.

In this example an attempt is made to reproduce the dynamic behavior of a DC torque motor. The observed input voltage and output angular position of the motor shaft is shown in the figure 1-1.

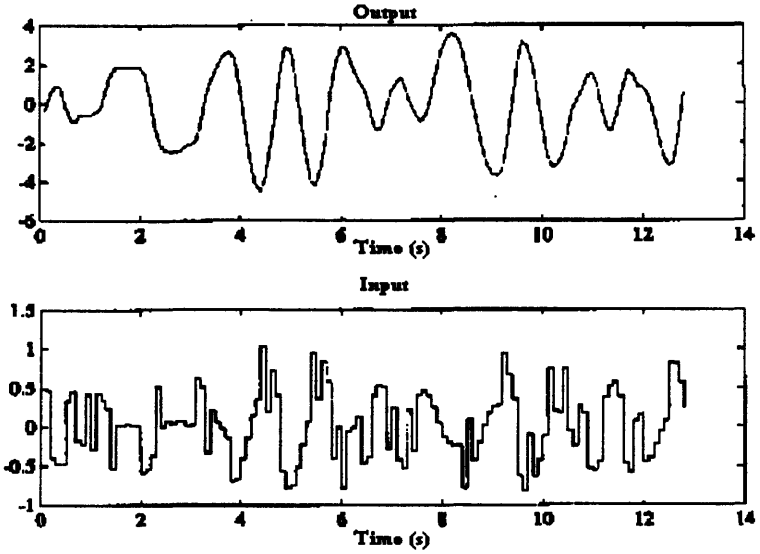

Figure 1-1: Motor input voltage (volts) and output angular position (rad) of the motor shaft.

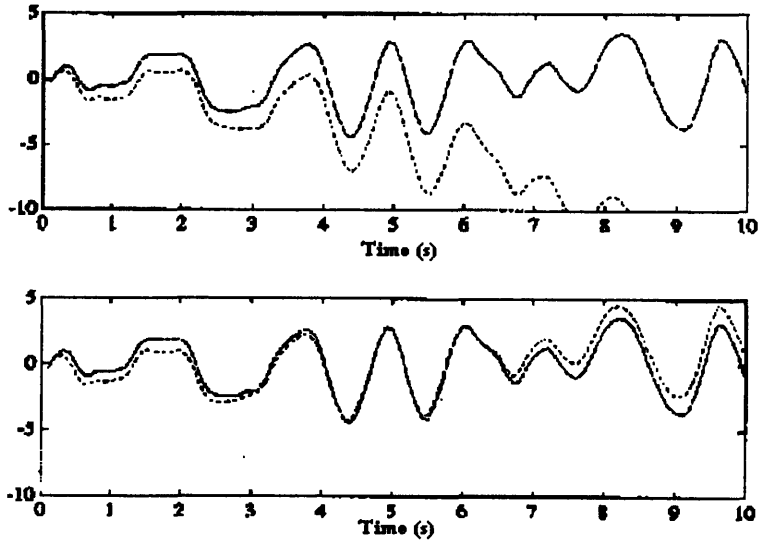

Figure 1-2: Measured output (solid line) and the simulated output (dashed line). The top diagram shows the simulated output of a data-driven model and the bottom diagram shows that of the physical-principle based model.

After processing the data, two models are created. One is made through the black-box

scheme and the other is based upon the physics of the DC motor. The two models are later used to predict the output response of the motor. The result is shown in the figure 1-2

From the results shown, the model made by the black-box approach shows an increasing drift that moves away from the actual response while the other model can produce a reasonably good match. The example seems to not only suggest that there is something deficient about the black-box approach but also implies that to reproduce the observed behavior, a physics-based model is the better choice. The result also suggests that the physics-based approach will give a good chance in getting the motor parameters. This is not to say that the black-box model cannot at all reproduce the observed behavior; besides drifting away, the model response in fact carries a resemblance of the actual response. What is going on here? To understand the problem better, this example of DC motor will be studied further in the thesis. To goal is to answer two questions: one, whether or not the black-box approach, which is purely data-driven, can provide a physical parameter estimate of a physical system and two, whether or not the physical-principle based approach, which incorporates the geometric and material properties of the physical system into the model, can reproduce the observed data and provide a good estimate of each physical parameter.

Before going into the details , it is the time to discuss the technical aspects of two modeling approaches used in the parameter identification in the thesis.

1.2 Organization of This Thesis

In the section 1.3 of this chapter, a physical model of a DC torque motor is introduced in a network representation. This physical network model will be used throughout the thesis as the basis for the physical-principle based approach.

Chapter 2: The overall procedure of "black-box" identification used in the thesis is introduced. The issues related to estimating the physical parameters from the discrete-time model parameters are discussed. A potential problem of non-unique mapping from behavior to physical parameters is also explained.

Chapter 3: The objective of this chapter is to apply both the "black-box" modeling and the physical principle-based modeling to synthesize a map from the simulated behaviors of a DC

motor to its physical parameters. The simulation is made to provide a disturbance-free data set. A physical network-based identification is also introduced.

Chapter 4: An experiment is conducted to identify the physical parameters of a DC torque motor. The identification methods in the previous chapter are also applied here.

Chapter 5: This final chapter summarizes the results found and also gives direction toward future works.

1.3 Network Models of Physical Systems

A physically parameterized model of a physical system can be represented by many means; the particular approach chosen here is through a network model. The advantage of using a network model is that it not only has an explicit representation of the physical parameters but also has a modular structure which is analogous to that of jigsaw pieces. When completely and correctly assembled, jigsaws pieces put on a meaningful display. The entire display can be spilt up into a number of jigsaws clusters each of which puts on a part of the entire display. In the same fashion, the physical network model, when correctly assembled, depicts an energy interaction within the physical system. A large network structure (*or port structure*) can be disassembled into a number of sub-structures under certain principles[1]. This modularity of the network model enables a divide-and-conquer approach to the identification of a complicated physical system. Essentially, a task of identification of a complex system could be broken down into a number of similar tasks of less complicated systems. Therefore, the algorithms developed for parametric identification of small subsystems may be used for larger complex systems. In this thesis, the focus will be the identification approach for a DC torque motor, which is a simple electro-mechanical system.

A network model of a DC torque motor is shown as a bond graph in the figure 1-3.

[1]For example, junction structures of a network need to satisfy power continuity condition [4]

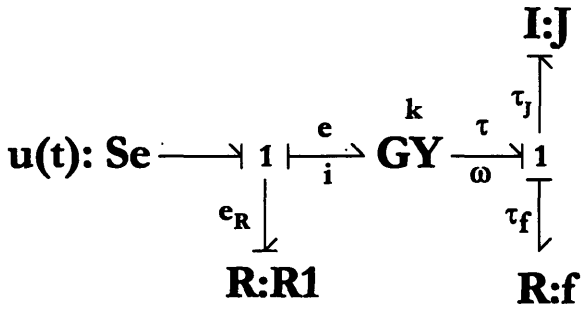

Figure 1-3: A network representation of a DC motor

From the network model, the problem now is to obtain the value of J, f, k, and R_1. The constitutive equations for the network model are as follows:

$$\begin{aligned} e_R &= iR_1 \\ \tau &= ki \\ e &= k\omega \\ \tau_J &= J\dot{\omega} \\ \tau_f &= f\omega \end{aligned}$$

If all the variables-$e_R, \tau, i, e, \omega, \dot{\omega}$, and τ_f-are measured then the four sought-for parameters will be known via the above constitutive equations. However, not all of the variables can be measured. For example, it is impractical to measure τ_f (damping torque) because, to solve for τ_f, requires the knowledge of τ and τ_J where $\tau_f = \tau - \tau_J$ from the continuity requirement at the 1 junction. However, both τ and τ_J depends on the unknown parameters k, and J, respectively. So, what seems like a straight-forward means of mapping from behaviors to the physical parameters is not applicable here. The next idea is to try to identify all parameters through the observable behaviors of the physical system. The approach taken here is first to select the observable behavior variables to be measured and then to construct a model which can adequately reproduce such behaviors. The final step is to estimate the physical parameters from the derived model. Two approaches of model construction will be explored:

one is to construct a model (or models) via the standard "black-box" system identification and the other is to construct the models with insights of the network structure of the physical system. To understand what the two approaches offer, a simulation of the behaviors of a DC torque motor is made and the results will be applied to the actual hardware for verification purposes. Before going further into the simulation, it is worth noting what is meant by the standard "black-box" system identification.

Chapter 2

System Identification

2.1 Estimation Problem

The identification problem confronted now is precisely an estimation problem which is the process of taking a set of measurements to identify quantities that are not readily available or cannot be measured directly. The quantities of interest in this context refer to the physical parameters in a network model. Generally, the goal of the estimation problem is to identify the parameters and model which best describes the observed data; therefore, the estimation techniques in the field of system identification may provide a mapping from the observed behaviors into the physical system parameters.

2.1.1 Ingredients of estimation problems

- Data set: This is a set of measurements taken from a physical system. The data set generally includes system inputs and outputs, which may be corrupted by measurement noise.
- A class of models: This is a class of candidate models of the observed system. Through an estimation process one of the models is identified as "best" model.
- Criterion of fit: This is a decision criterion in choosing what is the "best" model among a class of candidate models. A common criterion of fit is to minimize a function of the

predicted error (the difference between the true measurements and those predicted by the models); this function is often referred to as the loss function or cost function.

2.1.2 Parametric models

Since we want to relate the observed behaviors of a physical system to its parameters, the model should be parametric in nature. For the case of linear time invariant system, a common class of models refers to a class of parametric models of the form:

$$A(q)y(t) = \frac{B(q)}{D(q)}u(t) + \frac{C(q)}{F(q)}e(t) \tag{2.1}$$

where,

$$\begin{aligned}
A(q) &= 1 + a_1 q^{-1} + a_2 q^{-2} + ... + a_{n_a} q^{-n_a} \\
B(q) &= q^{-n_k}(b_1 + b_2 q^{-1} + ... + b_{n_b} q^{-n_b+1}) \\
C(q) &= 1 + c_1 q^{-1} + c_2 q^{-2} + ... + c_{n_c} q^{-n_c} \\
D(q) &= 1 + d_1 q^{-1} + d_2 q^{-2} + ... + d_{n_d} q^{-n_d} \\
F(q) &= 1 + f_1 q^{-1} + f_2 q^{-2} + ... + f_{n_f} q^{-n_f}
\end{aligned}$$

where $u(t)$ and $y(t)$ are the measured input and output to the system, respectively. $e(t)$ is a white noise sequence and q is defined as a time-shift operator : $q^{-1}y(t) = y(t-1)$.

The above model is parameterized by the unknown variables a_i, b_i, c_i, etc. The idea behind a "black-box" modeling is to choose appropriate model order n_a, n_b, n_c, etc and delay n_k and the estimates of a_i, b_i, c_i, etc that best reproduce the observed behaviors. Among the class of models 2.1, a very common family of models is ARMAX (Auto-Regressive, Moving Average, eXogenous model). The general form is as follows:

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \tag{2.2}$$

For the case, $C(q) = 0$, the resulting model is called ARX. An ARMAX model is a general-

ization of a linear regression model whose functional form is the following:

$$y(t) = \Phi(t)\theta + v(t) \tag{2.3}$$

where $y(t)$ is the measurement, Φ is a function of measurable quantities, θ is a vector of unknown parameters, and $v(t)$ is measurement noise. Note that the notation used here can represent either a scalar or vector quantity; the relationship is still valid. To illustrate a close relationship between an ARMAX model and a linear regression model, consider the following ARMAX model:

$$y(t) + a_1 y(t-1) + a_2 y(t-2) = b_1 u(t) + e(t) + c_1 e(t-1)$$

Such a model can be written in the linear regression form 2.3 where

$$\begin{aligned} \Phi(t) &= \begin{pmatrix} -y(t-1) & -y(t-2) & u(t) & e(t-1) \end{pmatrix} \\ \theta &= \begin{pmatrix} a_1 & a_2 & b_1 & c_1 \end{pmatrix}^T \\ v(t) &= e(t) \end{aligned}$$

The ARMAX model and linear regression model are the two classes of models that are frequently used in this thesis.

From the observed behaviors: y and u, the model of the form 2.1 is constructed. In the process of doing so, according to the "black-box" modeling paradigm, there are two main questions that need to be addressed. One is what order of A,B,C, etc are most appropriate to fit the observed behaviors? The other is what characteristics of the behaviors will provide sufficient information about the system dynamics to permit a good model reconstruction? In literature, there are answers to such questions. To identify the model order, the standard black-box approach suggests an educated guess via a number of test statistics such as Akaike Information Criterion (AIC), Minimum Description Length (MDL), and Final Prediction Error (FPE). For informative behaviors for model construction, a common practice is to use *persistently exciting* signals. The details of the statistics tests and informative behaviors are addressed in the next section.

2.1.3 Model order from data

To construct a parametric model of the form 2.1, one must identify the order of delay n_k, number of terms, n_a, n_b, n_c and so on. In order to estimate the order of the delay, a simple analysis, called correlation analysis, can be used.

Correlation Analysis

The idea behind correlation analysis is to estimate the impulse response of a system without using impulse as an input. Consider an LTI system with the impulse response h_k where

$$y(t) = \sum_{k=0}^{\infty} h_k u(t-k) + e(t) \tag{2.4}$$

where $u(t)$ is an stochastic signal with zero mean and auto-correlation function[1]: $R_u(\tau) = E[u(t)u(t-\tau)]$ where $E[u(t)]$ is the expected value of $u(t)$ and τ is *lag variable.* If the input signal is uncorrelated with the disturbances $e(t)$, the cross-correlation function between u and y is

$$\begin{aligned} R_{yu}(\tau) &= E[y(t)u(t-\tau)] \\ &= \sum_{k=0}^{\infty} h_k R_u(\tau - k) \end{aligned}$$

With an uncorrelated input (white), $R_u(\tau) = \lambda$ for $\tau = 0$ and zero otherwise. As a result,

$$R_{yu}(\tau) = \lambda h_\tau$$

In other words, the cross-correlation function between the input $u(t)$ and the output $y(t)$ is proportional to the impulse response of the system. If the input is not white, a common practice is to whiten the input by a filter L and also run the output through the same filter. The filter L is chosen to whiten the input as much as possible. By doing so, the cross correlation function between the filtered input and the filtered output is still proportional

[1]For the case of zero-mean signal, the term auto-covariance and auto-correlation means the same thing are often used interchangeably.

to the impulse response of the system. For a complete detail of the algorithm of correlation analysis, which includes the case of non-white input, see [7].

Therefore, the property of $R_{yu}(\tau)$ is then used to estimate the order of the delay from u to y. To illustrate how the $R_{yu}(\tau)$ is used, consider a hypothetical system:

$$y(t) = 3 * u(t - 10) \tag{2.5}$$

Assume the input $u(t)$ to be a zero-mean, white signal , the result of the correlation analysis is shown in figure 2-1.

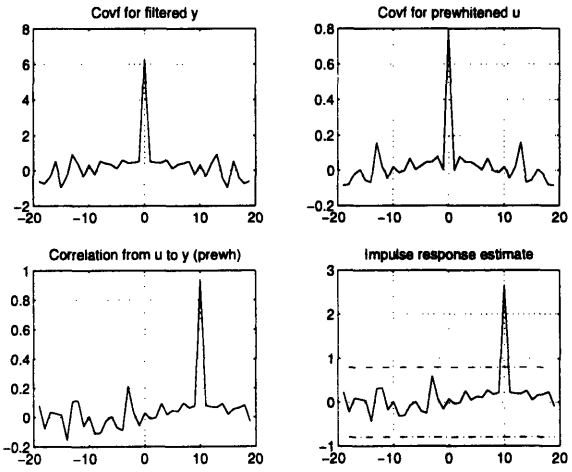

Figure 2-1: Correlation analysis of the system $y(t) = 3 * u(t - 10)$. The x-axis represents the value of lag variable.

Displayed on the figure 2-1 are (clockwise from upper left corner) the covariance function of prewhitened y, the covariance function of prewhitened u, an impulse estimate of the system, and the correlation from the prewhitened u to prewhitened y as a function of lag variable (x-axis). The dashed lines on the impulse response estimate represents 99 % confidence levels of the estimate. As mentioned earlier, in order to estimate the impulse response of a system by means of a correlation analysis, generally the input and output must be filtered through a whitening filter. The results of the filtered input and output are shown along with the estimate of R_{yu} and the corresponding impulse response for completeness. What we look for

in the correlation analysis is the amount of influence from u to y. For instance, significant correlation for positive values of lag variables corresponds to an significant influence from input u to later value of output y. Similarly, significant correlation for negative lag indicates a feedback from output y to input u.

From the correlation curve from u to y in the figure 2-1, it can be seen a distinctive peak at the lag = 10 which corresponds to the delay the hypothetical system. After determining the delay order, the next step in the black-box identification is to look for the "best" model structure.

2.1.4 Model structure determination

As the correct model order is not known a priori, it is only possible to postulate several different model structures, calculate them and then compare them. But how does one compare them? If a new data sequence (differing from that used in the identification process) is available, it is best to test each model against the new set of data and check the relative performance among different models. This new data set can only be used for the model comparison, no extra information can be extracted from it in model construction. If the model is to be compared against the data that is used in model construction, then a larger model will give a lower value of the loss function since it simply has higher degree of freedom to fit into the data. The extra parameters are basically used to fit to the specific disturbances present in the data set. This excess order presents the over-fitting problem and is not useful when the model is used with other kind of disturbances.

To get around this problem, the transition point must be found from the appropriate model to the over-fitted model. In literature, a number of different methods or tests exist and they all have the following characteristic:

$$min_{p,\hat{\theta}} \quad = \quad f(p,N)g(\hat{\sigma}^2(\hat{\theta}))$$

where $\hat{\sigma}^2(\hat{\theta})$ is the predicted error variance and $\hat{\theta}$ is the parameter estimates. The parameter p and N are the number of estimated parameters in the model and the number of data points, respectively. The function $f(p,N)$ increases with p and decreases with N while g is

a monotonically increasing function.

Such a test for model order determination will penalize a model that contains too many parameters by a minimization over the number of parameters p.

In the thesis, the following three well-known tests were used:

AIC (Akaike Information Criterion):

$$AIC(p) = log(\hat{\sigma}^2(\hat{\theta})) + 2p/N \tag{2.6}$$

$$p = \underset{p}{\text{argmin}}\, AIC(p)$$

MDL (Minimum Description Length):

$$MDL(p) = log(\hat{\sigma}^2(\hat{\theta})) + p/NlogN + p/Nlog(\hat{\theta}) \tag{2.7}$$

$$p = \underset{p}{\text{argmin}}\, MDL(p)$$

FPE (Akaike Final Prediction Error):

$$FPE(p) = \hat{\sigma}^2(1 + p/N) \tag{2.8}$$

$$p = \underset{p}{\text{argmin}}\, FPE(p)$$

where the symbol, $\underset{p}{\text{argmin}}$, represents the arguments of a function at the minimum value. Based from the numerical value of the three tests, the "best" model is chosen to be the one with both the lowest value of the statistics test and the least number of parameters.

In order to use these tests, a number of possible model structures are created from the same set of data and the corresponding values of the AIC, MDL, and FPE of each model are compared. The model that has the test value among the lowest and also has a minimum number of parameters is considered the "best" candidate.

Together with a good model structure, a successful identification requires an informative set of data which reveal the dynamics of the system of interest. The input signal is considered informative if it satisfies the condition of *persistence of excitation.*

Persistence of Excitation

An important component of an estimation problem is an informative data set, meaning that the data allows discrimination between any two different models in the set. The informative data is provided by a *persistently exciting* input:

Definition: A quasi-stationary signal $u(t)$ with spectrum $\Phi_u(\omega)$ is said to be persistently exciting if

$$\Phi_u(\omega) > 0, \qquad \forall\omega \tag{2.9}$$

For an exact definition of the spectrum, the reader is referred to [6]. The important thing to know is that the spectrum of a signal is the square of the absolute value of its Fourier transform; this is true for both continuous-time and discrete-time signals with finite energy. In the identification practice, the input spectrum $\Phi_u(\omega)$ needs to be non-zero over a wide range of frequencies to as great an extent as possible. An example of a persistently exciting signal is an independently, identically-distributed, white Gaussian signal; its spectrum is a non-zero constant at all frequencies.

2.1.5 Computing parameter estimates

With the model orders determined and an informative data set ready, the final step of "black-box" modeling is to identify the parameters in the models 2.1.

In searching for the "best" model, the criterion of fit is usually chosen to be the minimization of the mean-square error. This common choice of criterion together with an estimated model that is linear in parameters means that the parameter estimates can be found analytically by a linear least-squares approach. A common way to do this is to transform the standard linear model, such as an ARX model, into the form of linear regression 2.3. The parameter estimates then can be computed directly via pseudo-inverse:

$$\hat{\theta} \;=\; (\Phi^T\Phi)^{-1}\Phi^T y(t) \tag{2.10}$$

From equation 2.10, the results are the estimates of the model parameters that "best"

describe the observed behaviors. They shall be referred to as behavioral parameters since they are basically used to reproduce and predict the observed data. However, the geometric and material parameters are what we seek. Therefore, a relationship needs to be established from the behavioral parameters to the physical-system parameters. This is the topic of the next section.

2.2 Behavioral Parameters vs. Physical Parameters

From the previous discussion, one can think of the parameters of the model 2.1 as behavioral parameters; they describe the observed pattern of system behavior. Essentially, these parameters are for a sampled-data (discrete-time) model. What we seek, though, are the geometric and material parameters of the continuous-time model of a physical system. Therefore, to solve for physical parameters requires some way of transferring the parameters of a discrete-time model to those of an "equivalent" continuous-time model. In the literature, there are a number of ways to do so. A parameterized model such as ARX (and any other "standard" parameterized models in system identification) maps the discrete sequence of input data to the discrete sequence of output data; for instance, the ARX model maps input and output sequence through the following relationship:

$$\begin{aligned} A(q)y(t) &= B(q)u(t) + e(t) \\ A(q) &= 1 + a_1 q^{-1} + ... + a_{n_a} q^{-n_a} \\ B(q) &= q^{-n_k}(b_0 + b_1 q^{-1} + ... + b_{n_b} q^{-n_b}) \end{aligned}$$

where, $u(t)$, $y(t)$, and $e(t)$ are sampled input, sampled output, and disturbance, respectively. It is desirable to obtain the physical parameters of the system as they appear in the continuous-time model. For example, we would like to know the value of R_1, k, J, and f in a set of differential equations of a DC torque motor as shown below:

$$u(t) = R_1 i(t) + k\omega(t) \tag{2.11}$$

$$Jd\omega(t)/dt = ki(t) - f\omega(t) \tag{2.12}$$

where $\omega(t)$, $i(t)$, and $u(t)$ are angular velocity of the motor shaft, motor input current, and motor input voltage, respectively. To most people, the parameters a_i's and b_i's as appear in $A(q)$ and $B(q)$ in the above ARX model do not offer immediate insight as to what the physical parameters of its underlying physical system have to be. This difficulty immediately call for a means of transforming a discrete-time description such as ARX to its "equivalent" continuous-time model in the hopes of extracting information about the physical parameters from the parameters-a_i's and b_i's-which are used to capture system behaviors. There exist a number of the transformation schemes between the continuous-time model and its discrete equivalence; this is the topic of the next section.

2.2.1 Transformation tools

In the literature (for example, see [3]), there are a number of well-established transformation schemes between the continuous-time domain to the discrete-time domain. The most popular schemes include:*Backward rule* (or *Euler Approximation*), *Forward rule*, *Trapezoid rule* (or Tustin's formula), and *Zero-Order-Hold* (*ZOH*). Different techniques have different properties and result in different final forms of the transformed model. For the sake of concreteness of discussion, consider the set of differential equations 2.11 and 2.12 of a DC torque motor and find its discrete-time "equivalent" by applying the above transformations.

Choose the state variable to be the angular position of the motor shaft $\theta(t)$ and its angular velocity $\omega(t)$ and select the output measurement to be $\theta(t)$. We get

$$x(t) = \begin{bmatrix} \theta(t) \\ \omega(t) \end{bmatrix}$$

$$dx(t)/dt = Ax + Bu \tag{2.13}$$

$$y(t) = Cx(t) \tag{2.14}$$

$$\text{where} \quad A = \begin{bmatrix} 0 & 1 \\ 0 & -1/\tau \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ \beta/\tau \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

with $\tau = \frac{J}{f+k^2/R_1}$, and $\beta = \frac{k/R_1}{f+k^2/R_1}$. Let $\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \tau \\ \beta \end{bmatrix}$ We then can rewrite the above equations and obtain the following transfer function which maps the input voltage to the angular position of the motor shaft:

$$\begin{aligned} G(s,\theta) &= C(\theta)[sI - A(\theta)]^{-1}B(\theta) \\ &= \frac{\beta}{s(1+s\tau)} \end{aligned} \tag{2.15}$$

This continuous-time DC torque motor model will serve to illustrate the difference between the transformation schemes to the discrete-time domain. We shall consider two cases, starting from *Forward Rule, Backward Rule*, and *Trapezoid Rule* altogether due to their closely related approaches. Later the *Zero-Order-Hold* approximation is discussed.

Forward Rule, Backward Rule, and Trapezoid Rule

There are two means to carry out the Forward rule: one is to apply the Euler approximation directly to the state-space equation; the other is to apply the following change of variable $s = (z-1)/T$ to the continuous-time transfer function, $G(s)$, where T is the sampling period. Both result in the same final transformation if the model is linear. In the former case, for an arbitrary ODE

$$dx/dt = f(x,u)$$

The solution of this equation can be found by approximating $dx/dt = \frac{x_{k+1}-x_k}{T}$ where $x_k = x(kT)$ Therefore,

$$x_{n+1} = x_n + Tf(x_n, u_n)$$

Apply the Euler method to our state-space equation (1) and (2), we get

$$\begin{aligned} x[k+1] &= x[k] + TAx[k] + TBu[k] \\ y[k] &= Cx[k] \end{aligned}$$

Taking the z-transform on both sides:

$$\begin{aligned} zX(z) &= (I + AT)X(z) + TB(z)U(z) \\ X(z) &= T((z-1)I - TA)^{-1}B(z)U(z) \\ &= T\begin{bmatrix} z-1 & -T \\ 0 & z-1+T/\tau \end{bmatrix}^{-1} B(z)U(z) \\ X(z) &= T\begin{bmatrix} \frac{1}{z-1} & \frac{T}{(z-1)(z-1+T/\tau)} \\ 0 & \frac{1}{z-1+T/\tau} \end{bmatrix} B(z)U(z) \\ G_{fwd}(z) &= Y(z)/U(z) \\ G_{fwd}(z) &= \frac{\beta T^2}{(z-1)(T+\tau(z-1))} \end{aligned} \tag{2.16}$$

In fact, if the system is linear, we could have derived the above expression via a change of variable $s = (z-1)/T$ into $G(s)$ as follows:

$$\begin{aligned} G_{fwd}(z) &= \left[\frac{\beta}{s(1+\tau s)}\right]_{s=(z-1)/T} \\ &= \frac{\beta T^2}{\tau z^2 + (T-2\tau)z - (T-\tau)} \end{aligned} \tag{2.17}$$

Similarly, the Backward Rule and the Trapezoid Rule can be carried out by performing a change of variable $s = (1-z^{-1})/T$, and $s = \frac{2}{T}\frac{z-1}{z+1}$, respectively. Consequently,

$$G_{bwd}(z) = \left[\frac{\beta}{s(1+\tau s)}\right]_{s=(1-z^{-1})/T}$$

$$= \frac{\beta z^2 T^2}{(T+\tau)z^2 - (T+2\tau)z + \tau} \tag{2.18}$$

$$G_{trpzd}(z) = \left[\frac{\beta}{s(1+\tau s)}\right]_{s=\frac{2}{T}\frac{z-1}{z+1}}$$

$$= \frac{(\beta/2)T^2(z+1)^2}{(T+2\tau)z^2 - 4\tau z - (T-2\tau)} \tag{2.19}$$

The three transformation methods discussed in this section are based on the numerical integration methods (as indicated by their names). Using different methods, the derived discrete transfer functions have different stability properties. This can be easily seen by substituting $s = j\omega$ into each transformation rule $s = T(z)$ or more appropriately, $z = T^{-1}(s)$. Figure 2-2 shows the mapping of the left-half s-plane to the z-plane for the three transformation rules.

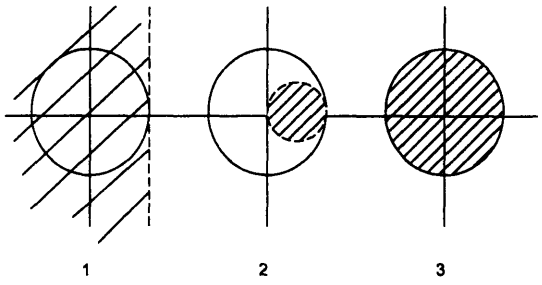

Figure 2-2: Stable s-plane poles get mapped into the shaded regions of the z-plane. The circle is a unit circle. 1. Forward rule 2. Backward rule 2. Trapezoid rule

Therefore, in order to maintain the stability properties of $G(s)$ the Euler method should be avoided.

Zero-Order-Hold Transformation (ZOH)

Consider again the differential equations 2.13 and 2.14 of the DC torque motor. This time, however, assume the values of the input are piecewise constant; i.e., $u(t) = u(kT)$

for $kT \leq t < (k+1)T$. This is the case, for example, in computer-controlled system, which is most likely in the experimental setting. We then have x(kT) and y(kT) given exactly by

$$x(kT+T) = A_T x(kT) + B_T u(kT) \tag{2.20}$$

$$y(kT) = G_T(z)u(kT) \tag{2.21}$$

$$\text{where, } A_T = e^{AT} \tag{2.22}$$

$$= \begin{bmatrix} 1 & \tau(1-e^{-T/\tau}) \\ 0 & e^{-T/\tau} \end{bmatrix}$$

$$B_T = \int_0^T e^{A\tau} B d\tau \tag{2.23}$$

$$= \begin{bmatrix} \beta(\tau e^{-T/\tau} - \tau + T) \\ \beta(1-e^{-T/\tau} \end{bmatrix}$$

$$G_{zoh}(z) = C(zI - A_T)^{-1} B_T \tag{2.24}$$

$$= \frac{\beta(\tau e^{-T/\tau} - \tau + T)(z - e^{-T/\tau}) + \tau\beta(1-e^{-T/\tau})^2}{(z-1)(z-e^{-T/\tau})}$$

$$G_{zoh}(z) = \frac{\tau\beta(T/\tau - 1 + e^{-T/\tau})z + (1 - e^{-T/\tau} - T/\tau e^{-T/\tau})}{(z-1)(z-e^{-T/\tau})} \tag{2.25}$$

$$G_{zoh}(z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{2.26}$$

$$b_1 = \beta\tau(T/\tau - 1 + e^{-T/\tau})$$

$$b_2 = \beta\tau(1 - e^{-T/\tau} - T/\tau e^{-T/\tau})$$

$$a_1 = -1 - e^{-T/\tau}$$

$$a_2 = e^{-T/\tau}$$

Notice that the original transfer function $G(s)$ contains poles at $s = 0$, and $s = -1/\tau$. The discrete-transfer function $G_{zoh}(z)$ has poles at $z = 1$, and $z = e^{-T/\tau}$. Therefore, the *ZOH* approximation maintains the dynamics of the original continuous-time model by mapping all the poles according to the well-defined relationship: $z = e^{sT}$. The stability property of the continuous-time model is still maintained in its discrete "equivalence". However, please

notice that the *ZOH* transformation also *introduces an additional zero* in the transfer function in the z-domain while none is present in the original transfer function in the s-domain.

From the previous discussion, it is quite obvious that the Euler approximation will not serve our purpose since it does not even guarantee the stability of the original system. The feasible ones are *Backward Rule*, *Trapezoid Rule*, and the *ZOH* . While the former two are considered as approximation methods, the last one is exact for the case of a sampled-data system. However, the *ZOH* transformation is also more complicated than both the *Trapezoid Rule* or the *Backward Rule*. So, the choices are there for the user to decide. In this thesis, the *ZOH* and the *Backward Rule* are used and their results are compared. *Trapezoid Rule* is also a possibility but it requires a little more mathematical manipulation than the *Backward Rule*. For the purpose of maintaining the stability of the transformed model, the simpler *Backward Rule* will work fine.

2.2.2 Non-unique map from behaviors to physical parameters

Consider a result of a transformation of the equation 2.11 and 2.12 (for instance, see the equation 2.26). It is apparent that the most one can get out of the behavior parameters, a_i's and b_i's are τ and β both of which are non-linear functions of the sought-for parameters: J, f, k, and R_1. This suggests that one may not be able to get a unique mapping from a particular choice of observed behavior to physical parameters. As seen in this case, the result of the identification is the estimates of τ and β where $\tau = \frac{J}{f+k^2/R_1}$, and $\beta = \frac{k/R_1}{f+k^2/R_1}$. As a result, there are infinitely many solutions of J, f, k, and R_1. To strive for a unique solution, an additional set of two equations is required. Hence, this demands for an additional choice of observed behaviors. Which behaviors should be observed? Whether one can get around the problem via data-driven approach ("black-box" identification) or physical-structure based approach is the issue explored in the next chapter.

2.3 Summary

In this chapter, the procedure for the "black-box" identification used in the thesis was introduced. It was found that the "black-box" approach produces a set of behavioral parameters

which need to be transformed into geometric and material parameters. The behavioral parameter is that of a discrete-time model while the sought-for physical parameter is for a continuous-time model. Such a mapping can be achieved by using a number of transformation techniques between the continuous-time model to discrete-time model. In this thesis, the chosen methods are the *Zero-Order-Hold* and the *Backward Rule*; both of which maintain model stability. After the transformation was carried out, it was found that a map from behavior to geometric and material parameters may be non-unique. That is, there are many possible values of physical parameters that can produce the same pattern of observed behavior. To find a unique mapping, the data-driven approach and the physical-structure based approach will be carried out. In the next chapter, a set of behaviors of a DC torque motor is simulated and the goal is to find out whether the two approaches can map the behaviors of the motor to its physical parameters.

Chapter 3

Physical Parameter Identification of A DC Torque Motor: Simulation

In this chapter, the two proposed approaches will be taken to identify the physical parameters of a DC torque motor. The first approach develops a model based on the black-box technique, the second on the knowledge of the network structure of the DC torque motor. Both methods will be applied to a set of simulated data. The simulation provides a controlled environment in which the data can be generated as cleanly as possible so that different results of the two modeling approaches are due solely to the nature of the approach. The results of this chapter will be further verified against experimental data in the next chapter.

3.1 Identification via Standard "Black-Box" Method

In this section, the "black-box" identification technique described in chapter 2 will be used to identify the physical parameters of a DC torque motor. The three components of the estimation problem needed are: the data set, a class of model, and a criterion of fit. The data set is chosen to be a pair of input and output variables. Without insights of physical structure, a choice of input and output measurements could be that of figure 1-1. That is, the voltage signal is the input and the motor shaft position the output. The class of model used is that of model 2.1 and the criterion of fit is minimizing mean squared-error. The goal in this section is to obtain the "best" model to reproduce the observed data and then try to

identify the physical parameters of the simulated DC motor from the derived model.

3.1.1 Simulation of a DC motor

In this section, a simulation of a DC torque motor similar to the example in the figure 1-1 is made. That is, the simulated input is the voltage signal and the corresponding output is the motor shaft position. From the figure, the recoded data used for the identification appears to be a random sequence. So, a random waveform is sought in the simulation. Based on the concept of persistence of excitation, the input signal is chosen to be an independently, identically-distributed Gaussian sequence.

Before going right into identifying the parameters of the simulated motor, it is interesting to see if our simulation can produce a similar kind of behavior as seen in figures 1-1 and 1-2.

In the DC motor example in chapter 1, a set of motor input voltage and motor position is recorded and a model of the following form is selected to reproduce the observed data.

$$y(t) + a_1 y(t-1) + a_2 y(t-2) = b_1 u(t-1) + b_2 u(t-2) \tag{3.1}$$

where u is the input voltage and y is the output motor shaft position. The above model certainly belongs to the class of model 2.1. It may not be the "best" model but it is certainly a valid model for the motor. A number of simulations of u and y are made with two major kinds of results. They are illustrated in the following two examples:

A DC Motor Simulation 1

In this example, the simulated input-output is shown in figure 3-1. It can be noticed that the magnitude of the simulated input is different from that in the figure 1-1. This difference should be immaterial since the simulated input is selected to be a white, Gaussian sequence which, according to the persistently exciting criteria, provides an informative set of data.

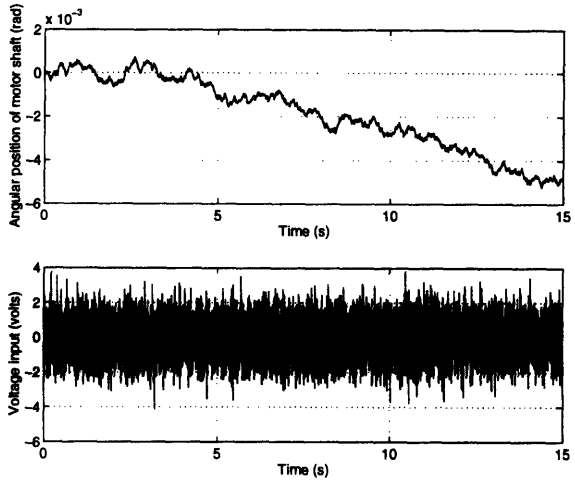

Figure 3-1: Simulated input-output time history of the DC torque motor. The input is a voltage to the motor (volts) and the output is the motor shaft position (rad).

The model 3.1 is derived from the above data set by solving a least squares fit. The model is then used to predict the same set of data that have been used at the model estimation. The result is shown in figure 3-2.

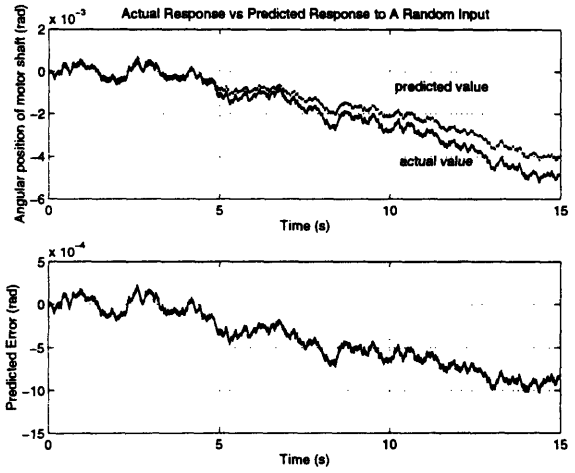

Figure 3-2: Actual (simulated) response versus predicted response of the model 3.1 to a set of white, Gaussian voltage input.

When tested against a different input signal, for example, a step input, the model predicts

the response as shown in the figure 3-3.

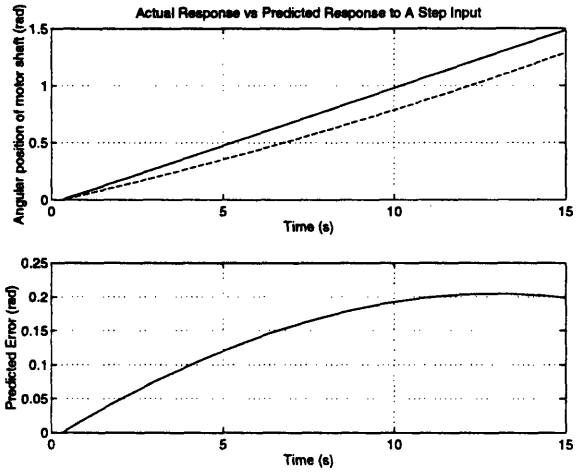

Figure 3-3: Actual response (solid line) versus predicted response (dashed line) to a step input.

From figure 3-2 and 3-3, the model does a reasonable job in reproducing the overall pattern of the motor behavior. However, based on the predicted step response (figure 3-3), it seems to indicate that the derived model is unstable. The predicted output (dashed line) of the model accelerates upward while the simulated response seems to increase linearly with time. Eventually the predicted output will exceed the simulated data. In fact, one of the poles of the model is located at $z = 1.0001$, which lies slightly outside the unit circle centered at the origin on the z-plane. This certainly indicates that the model is slightly unstable and agrees with the graphical appearance of the model's step response.

A DC Motor Simulation 2

In this second example, another set of white Gaussian signal is used as an input voltage to the motor. The corresponding time history of the input-output response is shown in the figure 3-4.

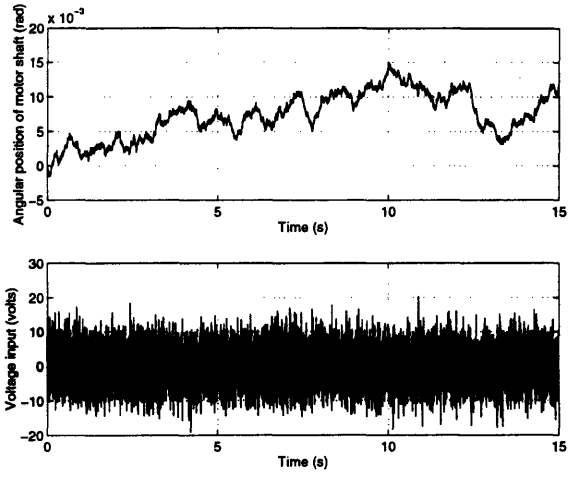

Figure 3-4: Simulated input-output time history of the DC torque motor. The input is a voltage to the motor (volts) and the output is the motor shaft position (rad).

With this new data set, a model of the form 3.1 is created by a similar manner as in the previous simulation. The derived model is used to reproduce the data in figure 3-4. The model is also tested against a step input. The results are shown in figures 3-5 and 3-6, respectively.

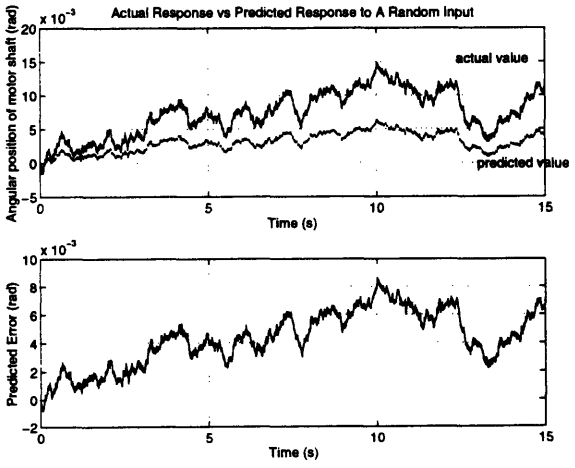

Figure 3-5: Actual (simulated) response versus predicted response of the model 3.1 to a set of white Gaussian voltage input.

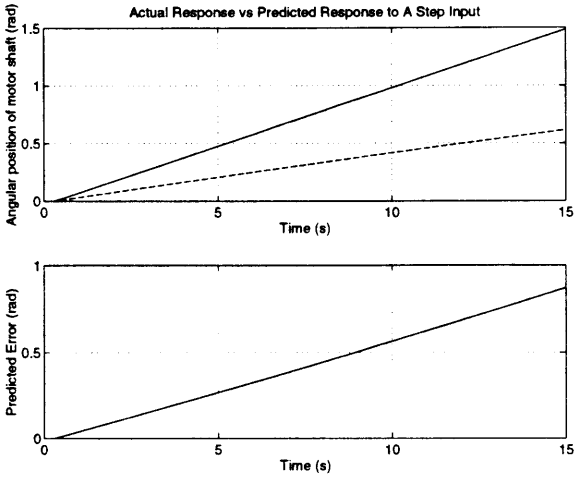

Figure 3-6: Actual response (solid line) versus predicted response (dashed line) to a step input

In this example, the model again can capture the overall pattern of the motor behavior though the error seems to grow with time. According to the model step response (see figure 3-6), the model seems to be stable; its response doesn't contain any sign of acceleration over the time course of simulation. In fact, the magnitude of both poles of the model is less than unity. One of the poles though, is at 0.999 which is at the edge of stability boundary.

The two examples clearly illustrate that there is a stability problem in constructing a model whose input is the voltage and output is the motor position. In one case, the derived model is unstable while, in the other case, it is stable. In both simulations, the models have one pole very close to 1. There is no guarantee which model one would end up with for this particular choice of observed behaviors.

The previous simulations are done in Matlab using a built-in function *lsim* which simulates an output of an ordinary differential equation. There are two algorithms implemented in *lsim*. In one, referred to as *first-order-hold*, the input data points are linearly interpolated while in the other algorithm, referred to as *zero-order-hold*, the input is held constant between each data point. In the previous simulations, the *first-order-hold* algorithm is used. For the sake of completeness, in appendix B, example 2 is repeated using the *zero-order-hold* algorithm. Basically, it was found that the chosen model structure 3.1 fits better to the data generated by the *zero-order-hold* algorithm.

In both simulations, the predicted responses from the models have a very similar pattern to that of the figure 1-2. That is, the models can reproduce the overall shape of the observed behavior but the predicted outputs drift away from the observed data over time. The similarity in behavior of our simulated motor and the associated models to that of the figures 1-1 and 1-2, while justifying the capability of the simulator, suggests a potential instability problem that could arise from the "black-box" approach.

In the next section, we take a step forward in attempting to identify the physical parameter from its behavior. In this attempt, we will follow the "black-box" identification procedure (described in chapter 2) to find the "best" model that reproduces the motor behavior. In the process of doing so, it is interesting to check if any instability problem arises and also to check for a unique identification of the motor's physical properties.

3.1.2 Instability problem and non-unique mapping in the "black-box" approach

In this section a new set of input voltage and the corresponding output motor position is generated and is shown in figure 3-7.

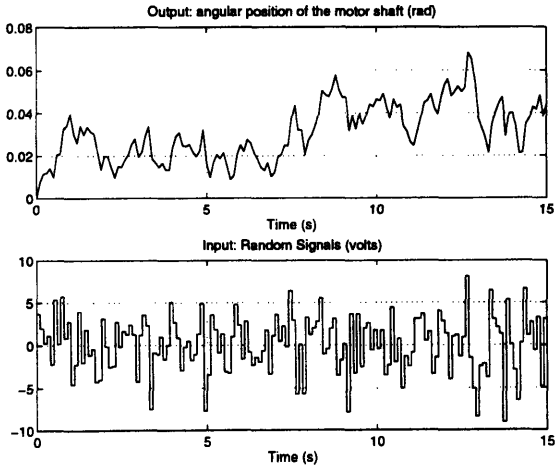

Figure 3-7: The input voltage (volts) and the corresponding motor position (rad). The input is a white, Gaussian sequence ($\mu = 0$, and $\lambda = 5$)

This data set is generated by the *zero-order-hold* algorithm. It was used here instead of the previous *first-order-hold* algorithm to make sure that the conclusion made is invariant

to the choice of the simulation algorithm. From the above, the first step is to test the order of the delay from the input signal to output signal; this can be done quickly by looking at the correlation from input to output.

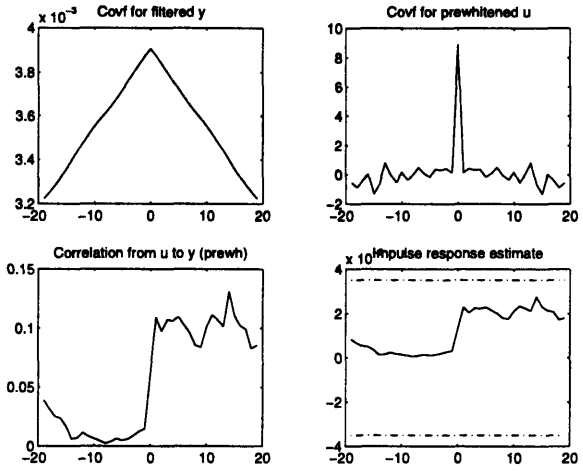

Figure 3-8: Correlation analysis between input voltage, u, and output angular position, y, of the motor shaft. The x-axis represents value of lag variable.

The correlation curve from u to y as a function of lag variable increases quite rapidly during the first non-negative lags. This indicates that the influence from the input to output is quite early as the lag = 0, 1, or 2. In this situation, lag value of 1 is chosen; that is the lag between the input u and the output y is of one sampling period. This order of delay is identical to that of 3.1.

After determining the order of delay, the next task is to find the order of the model 2.1; this can be done by examining the test statistics: AIC, MDL, and FPE. In the "black-box" approach, there is no prior knowledge of the physical hardware, so it is necessary to conjecture many model orders to be tested. The results are tabulated in table 3.1

Model Order			*Test Statistics*			
AR	MA	# of parameters (p)	AIC	MDL	FPE	Stable Model
1	1	2	-28.05	-28.01	65.50e-12	**No**
1	**2**	**3**	**-Inf**	**-Inf**	**0.00e-12**	**Yes**
1	3	4	-Inf	-Inf	0.00e-12	Yes
1	4	5	-Inf	-Inf	0.00e-12	Yes
2	**1**	**3**	**-Inf**	**-Inf**	**0.00e-12**	**Yes**
2	2	4	-40.29	-40.21	0.00e-12	Yes
2	3	5	-Inf	-Inf	0.00e-12	Yes
2	4	6	-Inf	-Inf	0.00e-12	Yes

Table 3.1: Test statistics for model order determination

The criterion of the three test statistics is to choose a "best" model to be that with the lowest value of test statistics and the least number of parameters. Hence, The three tests suggest that the "best" model is AR = 1 with MA = 2 and AR = 2 with MA = 1. The corresponding description of the two models are as follows:

$$y(t) + a_1 y(t-1) = b_1 u(t-1) + b_2 u(t-2) \tag{3.2}$$

$$y(t) + a_1 y(t-1) + a_2 y(t-2) = b_1 u(t-1) \tag{3.3}$$

The resulting parameters of model 3.2 are shown in table3.2.

Parameter	*Model 3.2*	*Model 3.3*
a_1	-1.000	-1.000
a_2	-	0.00e-4
b_1	20.19e-4	20.19e-4
b_2	0.01e-4	-

Table 3.2: Parameters of the model 3.2 and 3.3

As shown in the table, both models in fact are quite similar. Even though model 3.2 has one pole and model 3.3 has two poles, the value of a_2 of model 3.3 is 0.00e-4. Therefore, the output responses of the two models will be highly similar. Without loss of generality, the model 3.2 is chosen as the representative model.

From table 3.1, it can be noticed that most of the candidate models are found stable except one which is the model AR = 1 and MA = 1. It is obviously the "best" candidate model but the fact that it exhibits instability does agree with earlier results. As for the "best" model, it is found that one of its pole is in fact right on 1.00 which is the boundary of the stability region. This placement of pole very close to the stability boundary agrees with the results in the two example simulations. Therefore, it suggests that there is a chance that the derived model can be unstable like in simulation 1. Fortunately, in this case, the "best" model is marginally stable.

It is now of interest to see how well the derived model can reproduce the observed data. In addition to the data set used in model construction, there are two validating data sets: the unit step response and a sinusoidal response. The predicted responses are plotted against the actual simulated responses as shown in figures 3-9, 3-10, and 3-11.

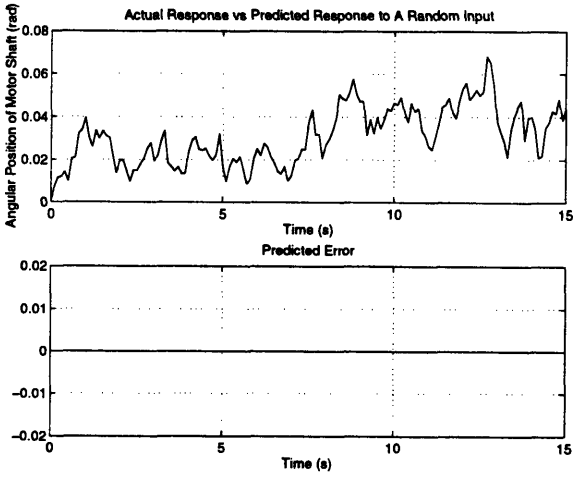

Figure 3-9: Predicted response of model 3.2 to a random input (zero-mean white Gaussian signal with variance = 5). The unit of the error shown is radian.

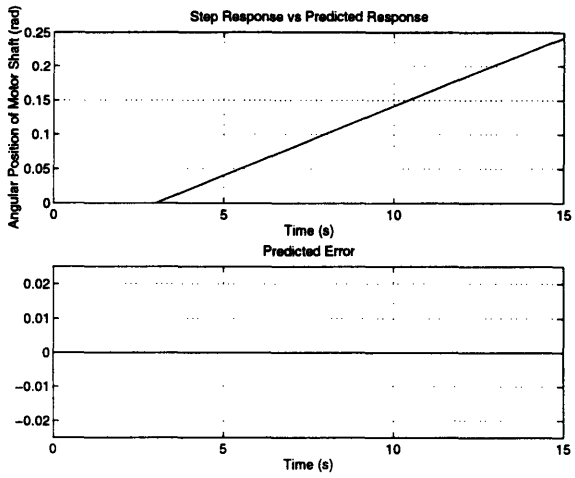

Figure 3-10: Predicted response of model 3.2 to a unit step input. The unit of the error shown is radian.

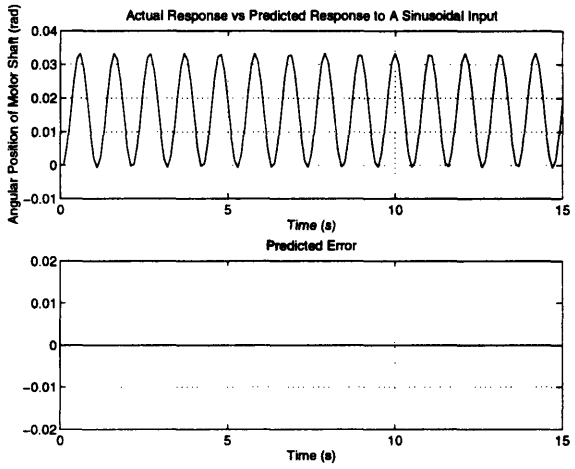

Figure 3-11: Predicted response of model 3.2 to a sinusoidal input: $u = 5sin(6t)$. The unit of the error shown is radian.

As seen from the plots, the model 3.2 predicts quite well in all cases. It is now left to check whether this model can serve as an adequate estimator of the physical parameters of the DC torque motor. In order to relate the model 3.2 to the physical parameter, a transformation from the the discrete-time model to the continuous time model (or vice versa) must be made. With the available network model, the differential equation relating the input voltage and

output shaft position is:

$$\ddot{y} + \frac{1}{\tau}\dot{y} = \frac{\beta}{\tau}u \tag{3.4}$$

where $y = \int \omega dt$ with $\tau = \frac{J}{f+k^2/R_1}$, and $\beta = \frac{k/R_1}{f+k^2/R_1}$. With the use of *zero-order-hold* transformation on the equation 3.4, the relationship between the parameters in the model 3.2 and the physical parameters in equation 3.4 is as follows:

$$\begin{aligned} b_1 &= \beta(T - \tau + \tau e^{-T/\tau}) \\ b_2 &= \beta(\tau - \tau e^{-T/\tau} - T e^{-T/\tau}) \\ a_1 &= -1 - e^{-T/\tau} \\ a_2 &= e^{-T/\tau} \end{aligned}$$

Through this relationship the estimated values of τ and β can be found and the results are compared against the actual value in table 3.3.

Parameter	*Actual value*	*Estimated value*	*Error*
τ	5.00e-5	0.00e-5	100.00 %
β	20.20e-2	20.19e-2	0.05 %

Table 3.3: Estimated values of $\tau = \frac{J}{f+k^2/R_1}$, and $\beta = \frac{k/R_1}{f+k^2/R_1}$ (simulation result).

Despite an excellent prediction of the motor behavior, the "best" model yields a very poor prediction of the physical parameter τ with 100 percent error. The large error in parameter estimation is not due the choice of the transformation, if for instance, the *Backward Rule* is used, the estimated τ and β are still numerically identical to that shown in the table 3.3.

Furthermore, even if the "best" model could result in a good estimate of τ, and β, they are still functions of the sought-for physical parameters: J, f, k, and R_1. It is now clear that, based on the estimated τ and β, getting an estimate of J, f, k, and R_1 is non-unique. There are only two equations with four unknowns. Two more equations are needed to uniquely identify all unknowns. What other additional behaviors should be observed to result in a

set of two more relations of the four parameters? There is no answer in the "black-box" approach and the user is left to find it in an ad-hoc manner.

3.1.3 Why the "black-box" approach is not adequate

It can be seen that a lot of problems occur in trying to synthesize a map from behavior of a DC motor to its geometric and material properties. The purpose of this section is to give some explanations as to why the approach is troublesome.

From the beginning of the black-box identification, the choice of behaviors on which the subsequent estimation problem is built can be represented by the network model shown in the figure 3-12.

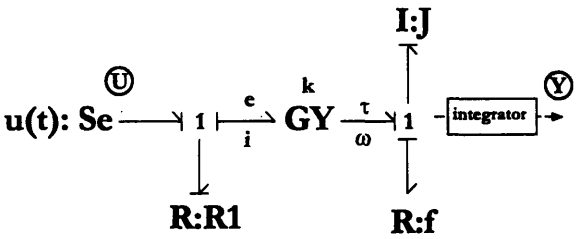

Figure 3-12: A selected pair of input voltage, u, and output shaft position, y, on a DC motor network model. The network contains a free integrator which determines the motor shaft position from velocity.

The output shaft position is obtained by integrating the shaft angular velocity which is determined by the motor inertia. The motor is driven by the command torque which is proportional to the amount of current in the motor winding. The magnitude of current results from the winding resistance subject to the control voltage source: $u(t)$.

As seen from the network representation, the selected choice of input-output contains a free integrator: $1/s$. The free integrator introduces a pole at the origin on the s plane which is mapped to the pole at 1 on the z-plane. By choosing to synthesize a mapping from the input voltage and the output motor position to the motor physical parameters, one is attempting to identify the free integrator. When one considers the network representation of the motor, one can immediately see that identifying the free integrator is useless; it provides no information about the geometric and material properties of the motor. The

order of the resulting model is higher than necessary. Moreover, attempting to identify a free integrator introduces a risk of obtaining an unstable model; examples are shown in the earlier sections in this chapter. Even when the simulation is conducted free of disturbances, some of the models (postulated as possible models) *turn out to be slightly unstable*; that is the identification scheme slightly over-estimates the magnitude of the pole. It is of interest to check what the result would be if the knowledge of network structure was used during the very first step in model construction and parameter estimation. For example, with the prior knowledge of the network structure of a DC torque motor, the choice of observed behaviors can be selected to avoid a free integrator which add an unnecessary order to a model. Furthermore, the physical network can perhaps give a guideline towards a complete set of observed behaviors to uniquely identify the physical parameters. The next section will focus on these issues.

3.2 Identification via Physical Structure: A Network-based Approach

In this section the four parameters of the DC torque motor will be identified with the knowledge of the network structure of the motor. A simple algorithm will be introduced which will take advantage of the property of the causal path on the network and its structure.

The algorithm for this network-based approach is as follows:

1. Construct a network-model of a physical system
2. On the network, choose a pair of behavior variables whose causal path does not consist of any free integrator. Denote the total number of physical parameters present on the network to be P. Then determine the following two causal paths:
   - From the chosen pair of variables, draw a complete causal path to determine what parameters will enter into the dynamics between the chosen variables. Denote the total number of such parameters to be p.
   - From the chosen pair of variables, draw a shortest causal path to determine the relative degree from one variable to the other. Denote this number to be R. Note

that the maximum relative degree is determined from the relative degree of the longest of all the shortest causal paths from any two possible variables present on the network model. Denote this number to be L.

3. Unless $2L - R + 1 = p = P$, seek an additional pair(or pairs) of behavior variables to be measured so that the total sum of $2L - R + 1$ for each input-output pair is equal to P.

4. Construct a dynamics equation describing each input-output pair and perform a backward-difference approximation on the equation. This establishes a set of P equations ($\sum(2L - R + 1) = P$) for the sought-for P physical parameters. The numbers R and L suggest the form of the discrete-time model used as an estimator (for each input-output pair) to be of the following form:

$$\underbrace{y(t) + a_1 y(t-1) + ... + a_L y(t-L)}_{L \text{ parameters}} = q^{-n_k} \underbrace{(b_1 u(t) + ... + b_{L-R+1} u(t-L+R))}_{L-R+1 \text{ parameters}}$$

The order of the delay n_k is determined via a correlation function from the input u to the output y. Now identify the parameters a_i's and b_i's that best reproduce the observed behaviors for each input-output pair.

5. Solve the resulting P algebraic equations for the P physical parameters.

In the proposed approach, the *Backward Rule* is used due to its simplicity and its reasonably good approximation (as seen its results as compared to that of *ZOH*).

At this point, it is of interest to apply the algorithm to identify the four unknown parameters: J, f, k, and R_1. With the knowledge of physical structure, the choice of observed voltage behaviors are chosen to avoid the presence of any free integrator in the dynamics. Therefore, one of the possibilities is to choose the input to be the voltage and the output to be the angular velocity of the shaft. The complete causal path and the shortest causal path are shown in figures 3-13 and 3-14, respectively.

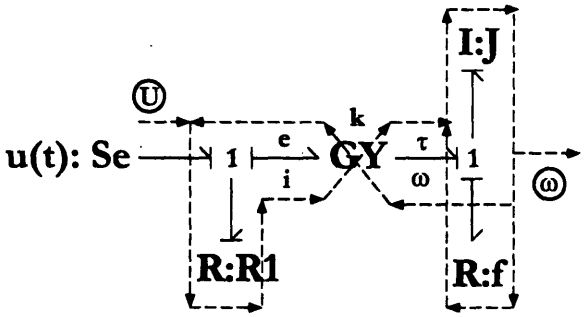

Figure 3-13: A complete causal path from the input voltage to the output shaft angular velocity

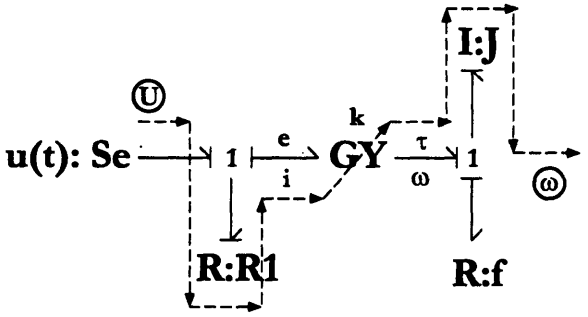

Figure 3-14: The shortest causal path from the input voltage to the output shaft angular velocity

From the network model and the causal paths, it is seen that $P = 4$, and $p = 4$. The largest relative degree $L = 1$ and the relative degree from the chosen input to the selected output is $R = 1$. This suggests an estimator for the observed behavior to be of the form:

$$\underbrace{y(t) + a_1 y(t-1)}_{L=1} = q^{-n_k} \underbrace{(b_1 u(t))}_{L-R+1=1}$$

where u and y are the voltage input and the motor shaft velocity, respectively. The delay, n_k, must be determined from the data set. From the previously determined causal paths, $2L - R + 1 = 2 < p = P$; as a result, another input-output pair is needed to identify all P parameters on the network. Since $p = 4 = P$, one is not restricted to choose a particular input-output pair. A possible choice is to choose an input current and output shaft velocity.

The causal paths for the new input-output pair is shown on the figure 3-15 and 3-16.

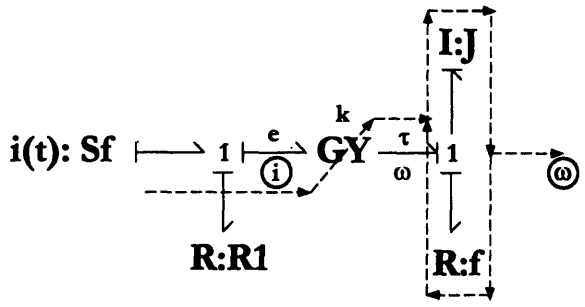

Figure 3-15: A complete causal path from the input current to the output shaft angular velocity

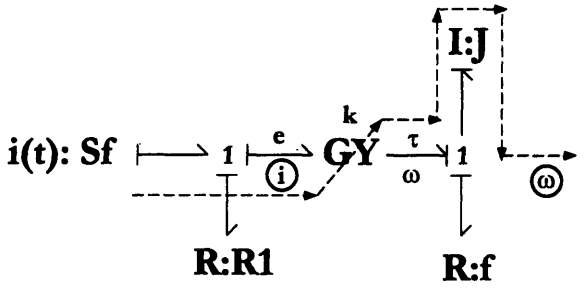

Figure 3-16: The shortest causal path from the input current to the output shaft angular velocity

With R and L determined to be 1, the form of the estimator is the same as in the previous first choice of input-output. Furthermore, it can be seen that the new input-output pair yields $p = 3$ and $2L - R + 1 = 2$. Therefore, with the measurements of both input-output pairs, the total sum of $2L - R + 1 = 4 = P$; this indicates that an estimate of all the physical parameters ($P = 4$) on the network can be found. With the choice of observed behaviors nailed down, the next task is to write the differential equations for both choices of input-output pairs and establish a relationship with the derived discrete-time models. The final approximation results a set of four equations for the four sought-for parameters: J, f, k, and R_1. The differential equation describing the chosen behaviors is the following:

$$\dot{y} + \frac{1}{J}(f + \frac{k^2}{R_1})y = \frac{k}{JR_1}u \tag{3.5}$$

Another input-output pair could also be the input current, i, and the output shaft velocity, $y = \omega$:

$$\dot{y} + \frac{f}{J}y = \frac{k}{J}i \tag{3.6}$$

With the equation 3.5 and 3.6 at hand, the corresponding "equivalent" discrete-time model of the two equations are: From the equation 3.5

$$y(t) + a_1 y(t-1) = b_1 u(t) \tag{3.7}$$

$$a_1 = -\frac{1}{1 + \frac{T}{J}(f + \frac{k^2}{R_1})} \tag{3.8}$$

$$b_1 = \frac{k}{JR_1} T / (1 + \frac{T}{J}(f + \frac{k^2}{R_1}) \tag{3.9}$$

From the equation 3.6

$$y(t) + a_2 y(t-1) = b_2 i(t) \tag{3.10}$$

$$a_2 = -\frac{1}{1 + f/JT} \tag{3.11}$$

$$b_2 = \frac{kT/J}{1 + f/JT} \tag{3.12}$$

The final step is to solve four algebraic equations for the four sought-for unknowns. From the equation 3.6 alone, with an accurate estimator one could achieve an adequate estimate of k/J, and f/J; but no information about R_1 is available. With the additional information from equation 3.5, one could estimate $\frac{k}{JR_1}$ and $\frac{1}{J}(f + \frac{k^2}{R_1})$. Therefore, an estimate of R_1 is obtained via $\frac{k}{R_1}$ and $\frac{k}{JR_1}$. Next an estimate of k is obtained through the quantity: $\frac{f}{J} + (\frac{k}{JR_1})k$ since $\frac{f}{J}$ and $\frac{k}{JR_1}$ are known from the model 3.7 and 3.10. With estimates of k and R_1 available, estimates of J, and then f immediately follow.

In the next section, the goal is to construct two estimators/models of the form 3.7 and 3.10 based on the measured data: input voltage, input current, and output shaft velocity.

3.2.1 A unique map from observed behaviors to physical parameters

The simulated input current and the corresponding output velocity are shown in figure 3-17.

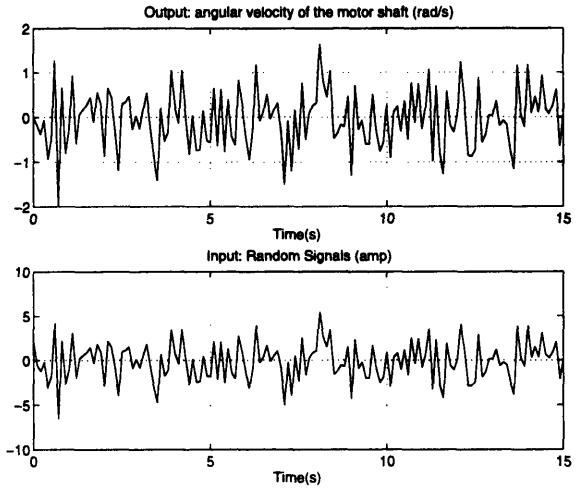

Figure 3-17: A simulation of white noise input current and the corresponding output angular velocity of the motor shaft.

Next, a correlation from the input current to the output velocity is determined and the result is shown in figure 3-18.

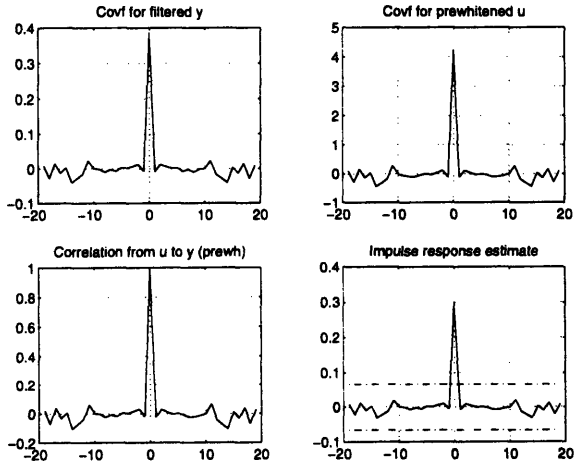

Figure 3-18: A correlation from the simulated input current to the output shaft velocity.

From the correlation curve, the influence from the input to the output is significant at zero lag. That is, there is no delay in the simulated data. Therefore, based on the previously proposed algorithm, the model for this set of data is of the form 3.10 whose parameters are determined by solving a linear least-squares fit to the data. The derived model is then validated against two different data sets: a unit step response and a sinusoidal response. The results are shown in the figure A-1 and figure A-2 in the appendix A. The derived parameters are shown in the table 3.4.

Model Parameters		*Physical Parameters*	
a_1	-52.97e-5	k/J	57.15e2
b_1	30.27e-2	f/J	18.87e3

Table 3.4: Estimated physical parameters of model 3.6

The next task is to construct an estimator for the model of the form 3.5. The time history of the input voltage and output angular velocity is shown on figure 3-19

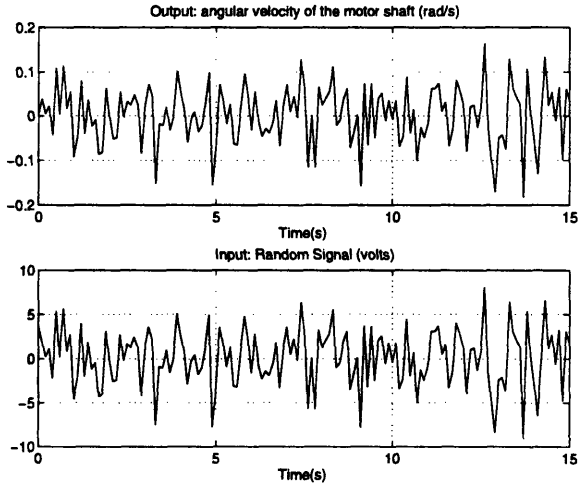

Figure 3-19: A simulation of white noise input voltage and the corresponding output angular velocity of the motor shaft.

Similar to what was previously done, a correlation curve is determined from the above

simulated data set and the result is very similar to that of figure 3-18. Hence, there is no delay in this set of simulated data either. Based on the previously proposed algorithm, the model for this set of data is of the form 3.7 and the derived model has the following parameters:

Model Parameters		*Physical Parameters*	
a_1	-52.66e-5	$\frac{1}{J}(f + \frac{k^2}{R_1})$	18.98e3
b_1	20.19e-3	$\frac{k}{JR_1}$	38.34e1

Table 3.5: Estimated physical parameters of model 3.5

From the estimated parameters in table 3.4 and table 3.5, the four sought-for parameters of the DC torque motor: J, f, k, and R_1 are found and tabulated in the table 3.6. As seen from the table, the estimated parameters are off by roughly five percent of the actual value.

Parameters	*Actual Value*	*Estimated Value*	*Percentage Error*
J $(N - m - s^2)$	52.95e-6	50.19e-6	5.21%
f $(N - m - s)$	1.00	94.71e-2	5.29%
k $(N - m/amp)$	30.29e-2	28.69e-2	5.28%
R_1 $(Ohms)$	14.90	14.90	0.00%

Table 3.6: Estimated physical parameters vs. the actual values used in the simulation

It can be seen in this chapter that a big problem of using the standard "black-box" approach to estimate the model is *the underlying structure of the model.* As previously seen, the '"black-box" identification will strive for the simplest possible model that can capture the observations and may result in an inadequate number of behavior parameters to provide an estimate of the physical parameters. Another problem which could arise is that with an input-output variable that consists of a free integrator in its dynamics, the resulting model is at risk of being unstable and may be unable to replicate the observed behaviors. Throughout the "black-box" approach, a "persistently exciting" signal was used to estimate

a model and, surprisingly, it was unable to give an accurate response, e.g., to a step input (figures 3-3 and 3-6). This result can attribute to the fact that the criteria of persistence of excitation only requires the signal spectrum to be non-zero over a wide range of frequency even though the system of interest may only have an energy highly concentrated at a single frequency. Therefore, a " persistently exciting signal" may not sufficiently excite the system at that particular frequency. As a result, the data set obtained may not be informative enough to synthesize an "accurate" model to the observed system.

With the physical structure, such as the network model, available, *the input and output variables can be carefully chosen* to prevent the presence of any free integrator in the observed dynamics (which are not useful at all in the identification of the physical parameter). Moreover, the network structure can provide a good estimate of a set of observed behaviors together with the associated model structure which are sufficient to result in a unique mapping to the physical parameters. It is important to note that the algorithm introduced here incorporates some of the features of the black-box identification. For example, the determination of the delay order. The network structure does not at all have this information built in. Only the observed data can show this.

Chapter 4

Physical Parameter Identification of A DC Torque Motor: Experiment

It is now of interest to test the earlier results obtained from the simulation and also to apply the identification scheme developed in the previous chapter to identify the physical parameters of an actual DC torque motor. The motor used is a brushed-type DC torque motor model number 2375V-096-149 manufactured by Vernitron Motion Control Group. The motor is, in fact, an integral part of the upper-arm amputation prosthesis emulator ([1], [2]). A brief specification of the motor as reported by the manufacturer is given in the table 4.1.

Parameters	*Units*	*Value*
Rotor Inertia	$N-m-s^2$	52.95e-6
Resistance	Ohms	14.900
Torque sensitivity	$N-m-s$	30.287e-2
Theoretical. No-Load Speed	rad/sec	120

Table 4.1: A brief specification of the DC motor 2375V-096-149

When the motor was put into operation, an aluminum hub and a steel shaft were installed onto the rotor. During normal operation, the motor shaft is attached to a pulley that turns the elbow part of the prosthesis emulator. With the addition of the hub and steel shaft, the

total inertia of the motor should be higher than that listed in the table 4.1. *In fact, a lower bound for the motor inertia in the setup was estimated to be 56.68e-06* Nms^2. The details of the calculation are in the appendix C.

4.1 Experimental Setup

In order to verify the results found in the previous chapter, the measurements of motor input current, voltage, motor shaft position and velocity are required. To obtain these measurements, a simple experimental setup was designed and is shown in figure4-1.

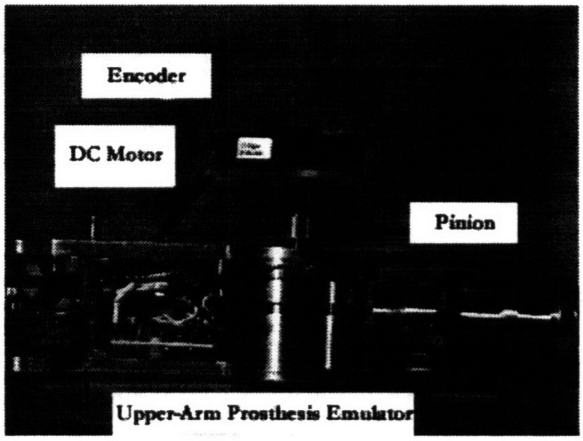

Figure 4-1: The experimental setup used in the physical parameter identification of the DC torque motor

The figure 4-1 shows the encoder mounted onto the motor shaft by a small pinion with two set screws. The DC motor is put into its housing in the prosthesis emulator. During the experiment, the motor is powered directly by an amplifier; the emulator circuitry is not involved in the experiment. During the experiment, the encoder is held by hand vertically and the pinion provides a good connection between the encoder shaft and the motor shaft.

The complete experimental setup consists of the following hardware:

- A brushed-type DC torque motor
- An amplifier as a voltage/current source: BOP 100-4M power supply produced by Kepco, Inc.

- An incremental encoder: model H1 with resolution of 2048 cycles per revolution. The manufacturer is US digital.
- A Data Acquisition Board: model AT-MIO-16E-2 produced by National Instrument.

In the experiment, the input and output measurements are sampled at the rate of 1000 samples per second. The amplifier, which provides the input signals to the motor, can be selected to operate in either voltage control mode or current control mode. In each mode, the amplifier provides a separate channel from which an amount of control input to the motor can be measured. The measurements of the motor shaft position is done by the encoder and motor velocity is then derived from differentiating the motor position. Hence, this setup provides the sought-for data needed to verify the results in the previous chapter.

In the next section, the identification procedures, both the "black-box" approach and the network-based approach, will be carried out to identify the the motor inertia, torque sensitivity, damping coefficient, and motor winding resistance.

4.2 Estimation via Standard "Black-box" Approach

The "black-box" identification (similar to that in the section 3.1.2) is repeated here with a set of real experimental data. The objective is to verify the results obtained in the simulation but with real data. In the simulation, a set of white, Gaussian signal is used to provide an estimation data set. In the experiment, however, a random white Gaussian input was avoided. The reason being that such input consists of many rapid and sudden changes in value which could cause damage to the motor brush. Instead,*a pseudo random binary input* was chosen as the data set used for identification. As in the simulation, the voltage signal to the motor is the control input and the motor shaft position is the observed output. An input-output time history is shown in figure 4-2.

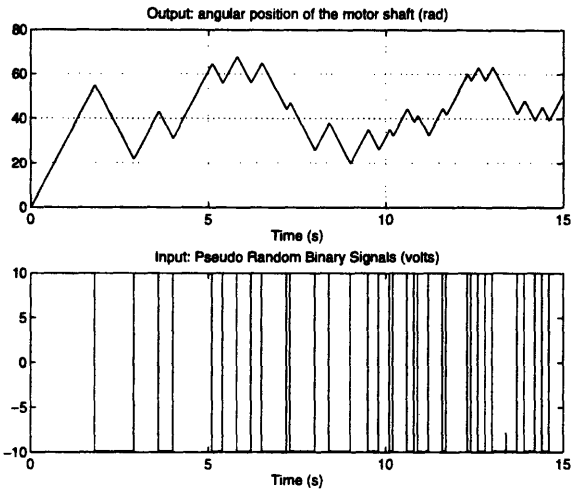

Figure 4-2: Experimental data of input voltage and output position of the DC motor. The input is a pseudo random binary sequence.

From this set of data, a correlation analysis was performed to estimate the lag from the input data, u, to the output data, y. The correlation curve is shown in figure 4-3.

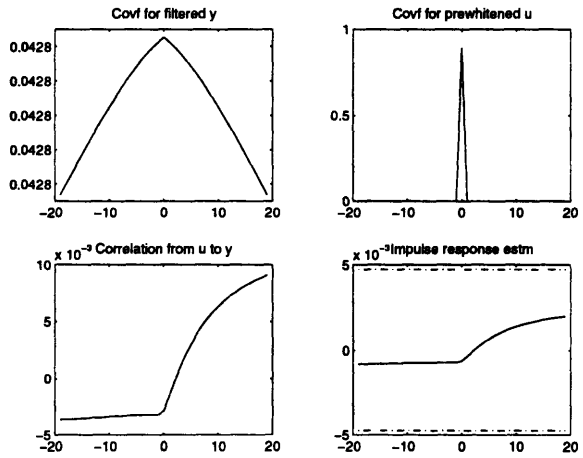

Figure 4-3: The correlation between the input voltage, u, to the output shaft position, y.

The correlation curve starts rising as the lag variable becomes positive and the correlation continues to increase. So it is unclear as to what order of delay dominates. In order to keep the identification procedure uniform with that in the simulation, the lag of 1 is chosen. So,

delay from u to y is one sampling period. This selection was then used to search for the "best" model structure. A number of model structures were postulated as before and the result of the statistic tests is shown in the table 4.2.

Model Order			*Test Statistics*			
AR	MA	# of parameters (p)	AIC	MDL	FPE	Stable Model
1	1	2	-14.47	-14.47	51.88e-8	No
1	2	3	-14.52	-14.52	49.21e-8	No
1	3	4	-14.57	-14.57	46.98e-8	No
1	4	5	-14.61	-14.61	45.26e-8	No
2	**1**	**3**	**-15.52**	**-15.52**	**18.22e-8**	**Yes**
2	2	4	-15.51	-15.50	18.46e-8	Yes
2	3	5	-15.49	-15.49	18.76e-8	Yes
2	4	6	-15.47	-15.47	19.06e-8	Yes

Table 4.2: Result of the statistics tests on the experimental data of input voltage, u, and output motor position, y.

A similar instability problem that appeared in the simulation of a DC torque motor also shows up here. As indicated in the table 4.2, some of the derived models are stable while some are not. In the experiment, more models are found to be unstable than that in the simulation. Errors in the measurements possibly contribute to this.

According to the tabulated results, the model that has the value of statistics test among the lowest and has the least number of parameters is considered the "best" candidate model. That model is AR=2, MA=1, which is the model of the form:

$$y(t) + a_1 y(t-1) + a_2 y(t-2) = b_1 u(t-1) \tag{4.1}$$

The structure of this model is, in fact, the same as that of the model which resulted from the statistics tests on the simulated data in section 3.1.2.

The parameters of the model 4.1 were calculated from the pseudo-binary input data set

and the derived model was then verified against the pseudo-binary input data set and two additional sets of data: a step input data and a sinusoidal input. The predicted responses, which appear as dashed-lines, are shown in the figure 4-4, 4-5,and 4-6.

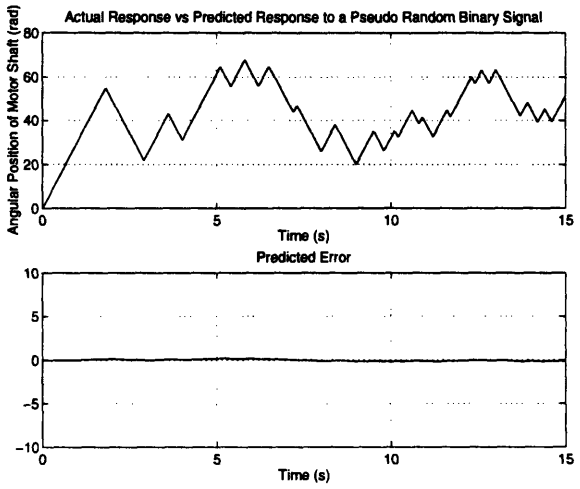

Figure 4-4: Measured output (solid line) vs. simulated output (dashed line) of the motor position using model 4.1. The input is a pseudo random binary sequence. The unit of the error shown is radian.

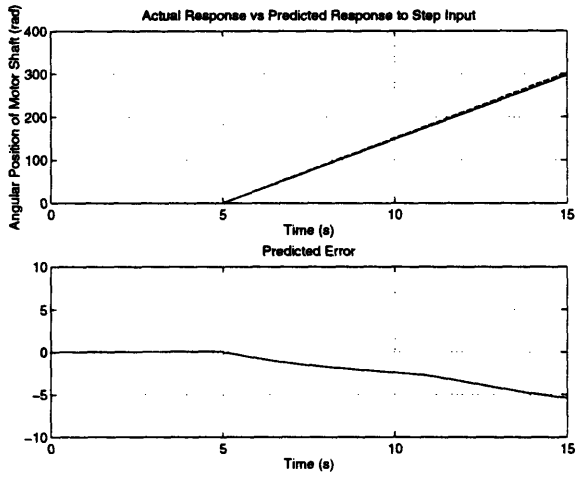

Figure 4-5: Measured output (solid line) vs. simulated output (dashed line) of the motor position using model 4.1. The input is a step signal. The unit of the error shown is radian.

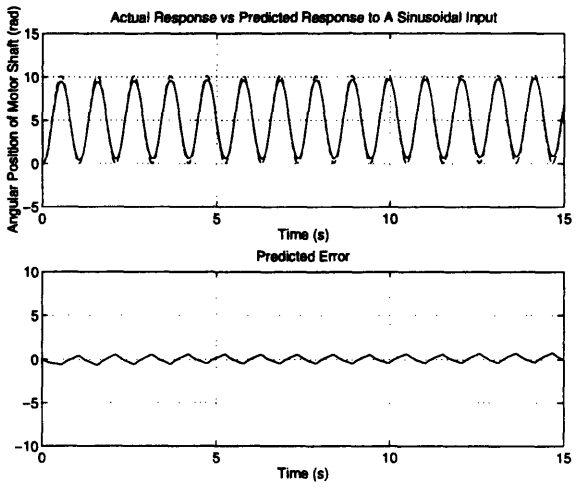

Figure 4-6: Measured output (solid line) vs simulated output (dashed line) of the motor position using model 4.1. The input is a sinusoidal signal. The unit of the error shown is radian.

The prediction from the model is generally good; over the time course of experiment, the largest error found is roughly 10 percent as shown in the sinusoidal response. The model predicts the shaft position to oscillate within the range from 0 radians to 10 radians but the actual output range is roughly 0.5 to 9.5 radians. As seen in the figure 4-6, the measured output position very slowly drifts upward as the motor oscillates back and forth over the time course of the experiment. This drift may result from the motor friction in one direction being higher than the other. The estimated drift rate is approximately 3.8 degree/second which is slow enough and should not pose a problem. Consider the figure 4-5, the derived model also over-estimates the step response of the motor by roughly 0.5 radians/sec; this results in a more or less straight line prediction error with a slope of -0.5 radians/second. As to the response to the pseudo-random input, the derived model performs quite well.

The "best" model found is marginally stable and its derived parameters are shown in tables 4.3.

Parameter	*Model 3.2*
a_1	-1.892
a_2	0.892
b_1	32.86e-5

Table 4.3: Parameters of the model 3.2

In this case, both poles of the model are within the stability region with one of them located right on the border line. As previously seen in the simulation, the discrete-time parameters in table 4.3 can be used to estimate physical parameters τ and β both of which are non-linear functions of the sought-for parameters: J, f, k, and R_1. Since an estimate of f is unknown, the estimated error of τ and β cannot be calculated. However, the estimated values of τ and β are shown here for completeness.

Parameter	*Estimated value*	*Unit*
τ	82.67e-4	*1/sec*
β	30.44e-1	*sec/volts*

Table 4.4: Estimated values of $\tau = \frac{J}{f+k^2/R_1}$, and $\beta = \frac{k/R_1}{f+k^2/R_1}$

Thus far, it can be seen in this section that the conclusions drawn from the results of the "black-box" identification in the simulation (see section 3.1.2) are valid in the experiment. The standard "black-box" approach could produce a stable model that can replicate system behavior quite well. However, the approach could also result in an unstable model. Together with the problem of model instability, the problem of non-unique mapping from the observed behavior to the physical parameters still remains. Both τ and β are non-linear functions of four parameters. Knowing how the motor position behaves as a function of input voltage is not adequate and the data-driven approach provide no means of selecting additional kinds of behaviors to unwind a unique map to the physical parameters. This example is just another demonstration that knowing a particular choice of behaviors can't guarantee a unique

identification of each physical parameter that produces such behaviors.

4.3 Estimation via Physical Network-based Approach

In this section, the network-based algorithm introduced in chapter 3 will be verified against the experimental data. The network approach suggests the measurement of the motor voltage input, current input, and the output angular velocity of the motor shaft. In the experiment, only the encoder is used to measure the angular position of the motor shaft from which the shaft velocity is derived by first-order finite differencing the shaft position. The derived velocity is filtered through a fifth order butterworth filter with cut off frequency at 100 Hz. The filter can get rid of most of the quantization error. After filtering in the forward direction, the filtered velocity sequence is then reversed and run back through the filter. This scheme provides the filtered velocity with zero-phase distortion.

Based on the same network model of the DC motor the idea now is to construct a model from the input voltage and output motor shaft velocity. The random white Gaussian input is avoided in the experiment for the same reason as previously explained. So instead, a pseudo random binary input is chosen as the data set used for identification. The resulting input-output time history is shown in the figure 4-7

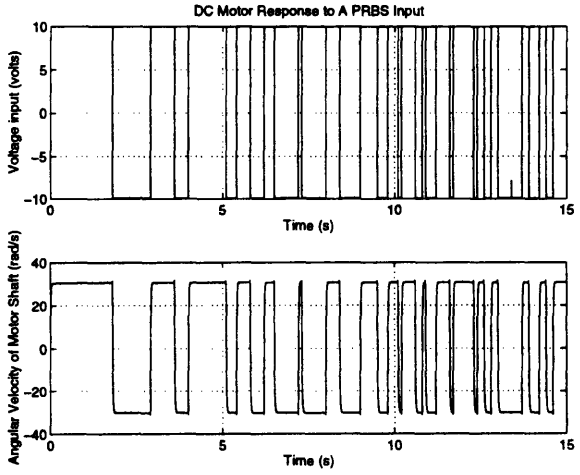

Figure 4-7: Experimental data of input voltage and corresponding output motor velocity. The input is a pseudo random binary sequence.

The input voltage assumes the value either -10 volts or +10 volts. As seen in the plot, the measurements of voltage is quite clean. A slight bias of the input voltage at 10 volts is noticeable but doesn't pose a serious problem. From this set of data, the next step is to look for any delay between the input and the output data. The amount of delay can be determined by performing a correlation between the control input and observed output data. The correlation curve is shown in the figure 4-8.

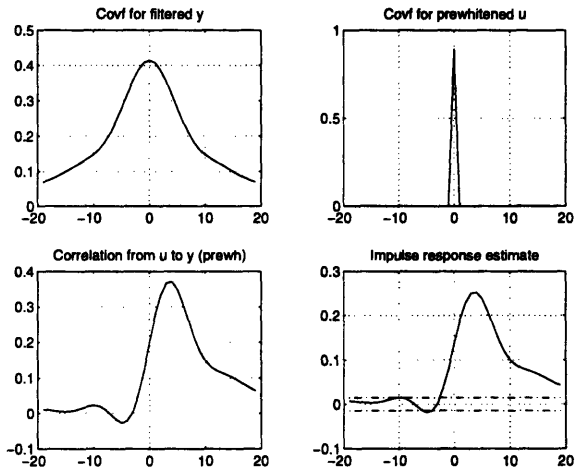

Figure 4-8: The correlation between the input voltage, u, to the output shaft velocity, y. The x-axis represents value of lag variable.

From the correlation curve from input voltage, u, to the motor velocity, y, the influence from u to y is most significant at the lag variable of 4 and the corresponding level of correlation is approximately 0.36. However, more than half of that level of correlation already appears starting at the lag of 0. So, a number of values of lag variable are tried in model construction and according to the identification algorithm in the section 3.2, the following model structure is suggested from examining the causal path from the input voltage and output velocity:

$$y(t) + a_1 y(t-1) = q^{-n_k}(b_1 u(t)) \tag{4.2}$$

where n_k, assumes the value of from 0 to 4. It turns out that, with different values of n_k,

the estimated values of a_1, b_1 are approximately the same (the numerical values are shown later); however, the model with $n_k = 1$ reproduces the observed data just a little better and it is, therefore, chosen as a representative model. A sample of the model performance is shown in figure 4-9.

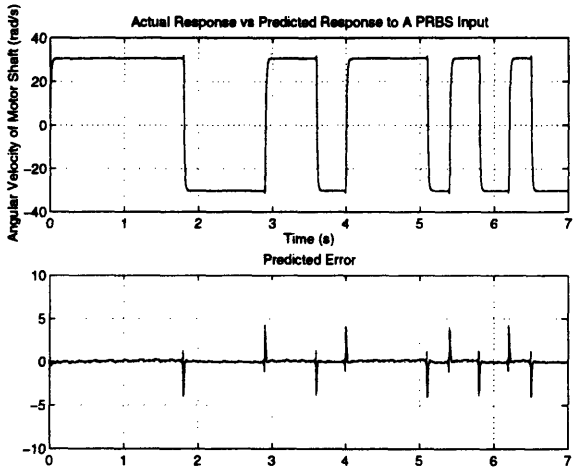

Figure 4-9: Measured output (solid line) vs. predicted output (dashed line) of the motor velocity using model 4.2. The unit of the error shown is rad/sec.

At this stage, the model provides two behavior parameters, a_1 and b_1 for the four unknown physical parameters. So, the next step is to look at different type of behavior so that the resulting model will yield an additional two behavior parameters. Based upon the motor network model (see figure 3-13), the complete causal path from the input voltage, u to output shaft velocity y, includes all the four sought-for physical parameters: J, f, k, and R_1. Therefore, according to proposed network approach (3.2), the next choice of observed behavior pair needs not have a complete casual path through all four parameters. What we are after is merely two additional behavior parameters. This is the purpose of the next experiment.

Examining the network model of the motor indicates that using current as a control input and the motor shaft velocity as an output will give exactly two more behavioral parameters (see figure 3-15 and 3-16). The complete causal path between the input and output doesn't carry any information about R_1 but it is of no concern.

To begin the second set of experiments, the amplifier is switched to operate in the current mode and the angular velocity of the motor shaft is determined. The data set chosen for identification is a pseudo random sequence for the safety of the motor brush. An input-output time history is shown in the figure 4-10.

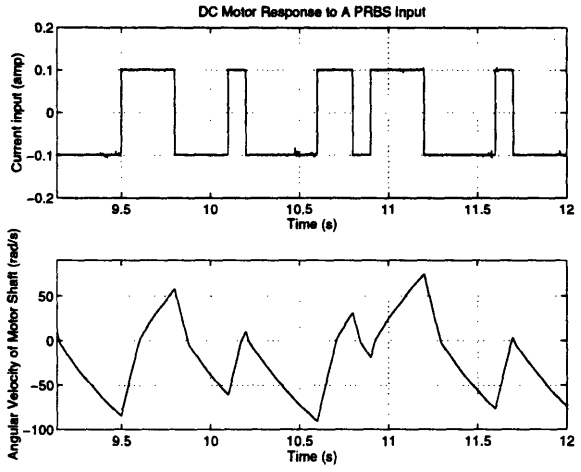

Figure 4-10: Experimental data of input current and corresponding output motor velocity. The input is a pseudo random binary sequence.

The data set obtained from the current channel shows that a certain portion of the data contains more irregularities than the others. To achieve a good identification of the motor dynamics a cleaner portion of the data is used to create a model. That is what is shown here in figure 4-10. As seen in the figure, the input current contains small disturbances which appear as small intermittent spikes along the signal waveform every now and then. The disturbances can be caused by either small noise in the input current channel or a quantization error during the conversion of the input current. On the figure 4-10, there is visible evidence showing a slight non-smooth transition as the motor changes its direction of rotation. The rate of change of the motor velocity changes slightly as it goes through zero. This behavior can be attributed to a higher motor friction in one direction than in the other. This finding is consistent with that found earlier in figure 4-6. The network model doesn't have this property but the network model should serve as a decent approximation.

From the data set shown in figure 4-10, a correlation analysis was performed to look for a

possible delay between the input and output signal. The result is shown in the figure 4-11.

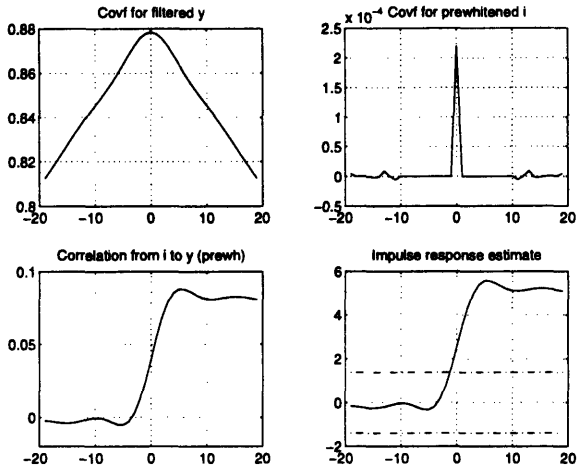

Figure 4-11: The correlation between the input current, i, to the output shaft velocity, y. The x-axis represents the value of lag variable.

From the correlation curve from input current, i, to the motor velocity, y, the influence from i to y is most significant at the lag variable of 5. However, level of correlation already appears starting at the lag of 0. So, a number of values of lag variable are tried in model construction. Based upon the causal path from the current to the output velocity (see figure 3-15 and 3-16), the model of the following form is created:

$$y(t) + a_2 y(t-1) = q^{-n_k}(b_2 i(t)) \tag{4.3}$$

where i is the control current input and y is the motor velocity. The order of delay n_k, assumes the value from 0 to 5. It turns out that, with different values of n_k, the results of the model parameters are quite similar. Without significant difference in result, the model with $n_k = 1$ is chosen as a representative model. The result of the model prediction to the random current signal is shown in the figure 4-12.

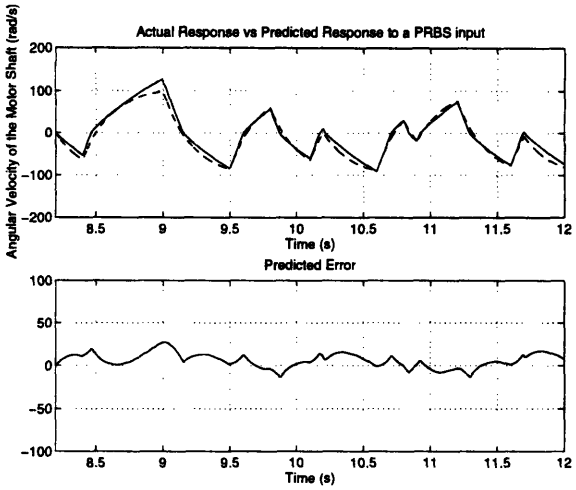

Figure 4-12: Measured output (solid line) vs. predicted output (dashed line) of the motor velocity using model 4.3. The unit of the error shown is rad/sec.

From the figure, the derived model results in a larger predicted error than that in the figure 4-9. In addition, the model predicts a smooth transition as the motor changes the direction of rotation while in a real data such a transition is not smooth at all. The large error in prediction may result from the fact that the relatively clean data segment from the current channel is quite limited. In the data shown here, only the data portion from the 8th second to the 12th second is used in constructing the model. If a larger and less noisy data set is used, the model prediction could be improved. Despite a large error, the derived model can reproduce the overall pattern of the motor velocity quite well. At the moment, we shall use this model anyway and proceed to identify the physical parameters that appeared in the network model of the motor.

At this point, there are enough behavior parameters to solve for the four motor parameters; the problem is simply solving a set of four algebraic equations for the four unknowns. The resulting estimates of the behavior parameters: a_i's and b_i's together with the estimates of the four physical parameters are shown in the table 4.5.

Model Parameters		*Physical Parameters*		Percentage Error
a_1	-90.38-02	J	63.04e-06 $(N - m/s^2)$	11.2 %
b_1	29.15e-02	f	30.729e-05 $(N - m/s)$	Not available
a_2	-99.52e-02	k	31.50e-02 $(N - m/amp)$	4.01 %
b_2	49.71e-1	R_1	15.49 $(Ohms)$	3.96 %

Table 4.5: Estimated parameters of the DC torque motor. The information about the motor friction is not available for comparison so it is omitted here.

The percentage error are calculated from the difference in the estimated physical parameters and the corresponding parameters shown in the table 4.1. Note that, besides the percentage error in inertia, all of the percentage error found are within 5 percent; the results agree with that in the simulation. To calculate for the percentage error in motor inertia, a rough estimate for the lower bound of the motor inertia has to be made to compare with the result from the identification process; this may contribute to some of the error found.

With the estimated parameters available, it is of interest to find out if the network model equipped with the estimated parameters can reproduce the motor behaviors. In a sense, this exercise can also be used to verify the estimated friction coefficient,

In what follows, the network-model of the motor is used to simulate the motor responses to three different input signals: a pseudo-random binary input, a step input, and a sinusoidal input. Figures 4-13 to 4-16 show the predicted motor position and velocity in response to a pseudo-random binary input and a step input.

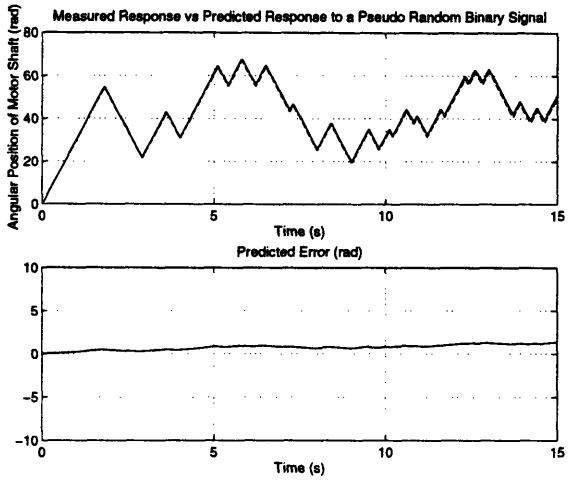

Figure 4-13: Measured output position(solid) and the simulated output position(dashed) of the DC motor shaft in respond to a set of pseudo-random binary input. The simulation is made from the network model using the estimated physical parameters. The unit of the error shown is radian.

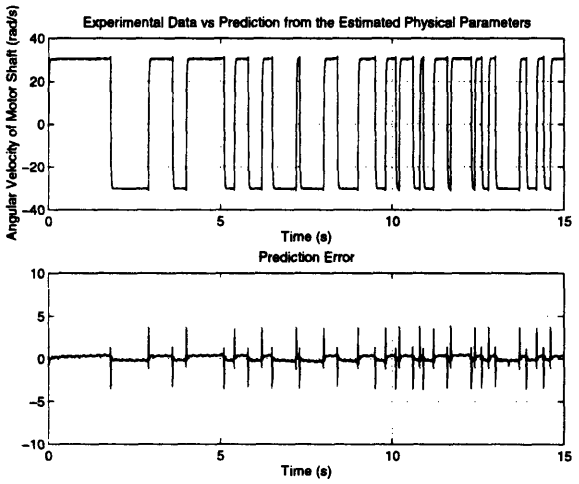

Figure 4-14: Measured output velocity(solid) and the simulated output velocity(dashed) of the DC motor shaft in respond to a set of pseudo-random binary input. The simulation is made from the network model using the estimated physical parameters. The unit of the error shown is rad/sec.

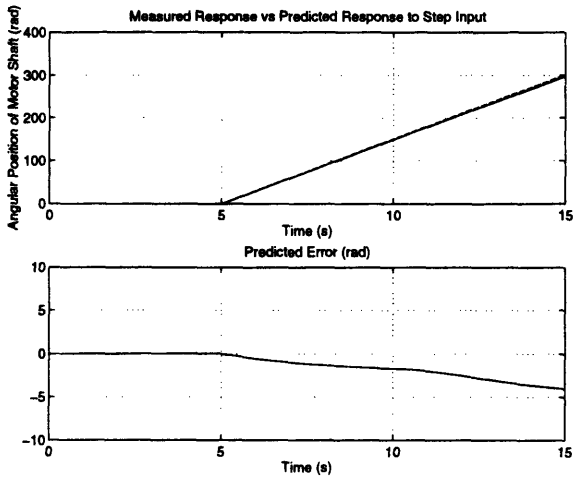

Figure 4-15: Measured output position(solid) and the simulated output position(dashed) of the DC motor shaft in respond to step input. The simulation is made from the network model using the estimated physical parameters. The unit of the error shown is radian.

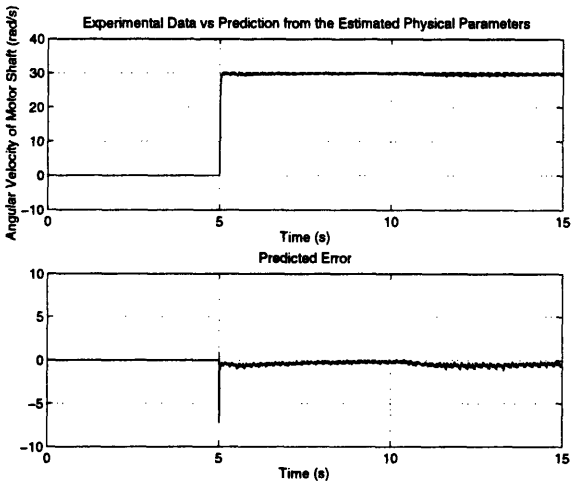

Figure 4-16: Measured output velocity(solid) and the simulated output velocity(dashed) of the DC motor shaft in respond to step input. The simulation is made from the network model using the estimated physical parameters. The unit of the error shown is rad/sec.

Overall, the predictions generated by the set of estimated physical parameters are quite good. For instance, in predicting the motor speed in respond to a pseudo-random binary

voltage (see figure 4-14), the network model can successfully reproduce the general pattern of the motor speed with some error during the sudden change in velocity. The plot of motor position (figure 4-13) shows a slight increase in predicted error over time; this attributes to a higher motor friction in one direction than in the other. In respond to a step signal, the model predicts the motor to rotate at a fixed angular velocity, approximately 30 rad/sec in this case. The actual response though shows a more or less constant velocity with some small "chattering". As we examine closer (see figure 4-17, it becomes clear that the actual motor speed is in fact varied between 29 rad/sec and 30 rad/sec.

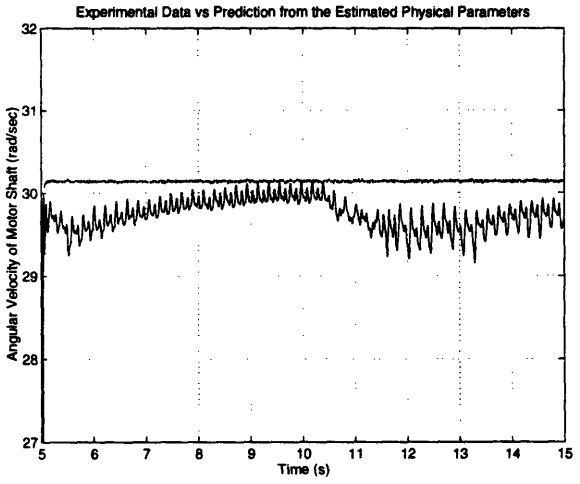

Figure 4-17: A magnified view of the measured output(solid) and the simulated output(dashed) of the DC motor in respond to a step input. The simulation is made from the network model using the estimated physical parameters.

Some quantization noise, resulting from a first order finite difference of the motor position, may contribute to some of the "chattering". The chattering may also result in part from the fact that the motor slightly accelerated and decelerated over the time course of the experiment. The overall speed is always within 29-30 rad/second which is close to what the model predicts. As for the response to a sinusoidal signal, the model performs reasonably well as seen in the figure 4-19.

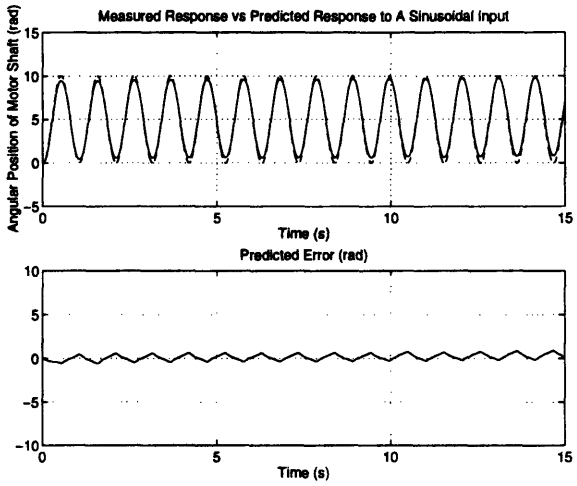

Figure 4-18: Measured output position(solid) and the simulated output position(dashed) of the DC motor shaft in respond to a sinusoidal input. The simulation is made from the network model using the estimated physical parameters. The unit of the error shown is radian.

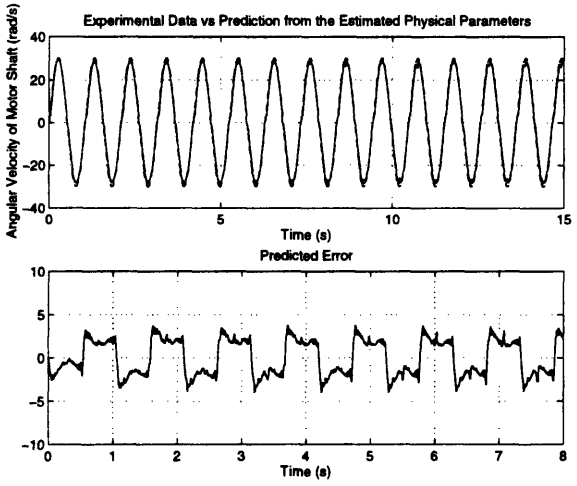

Figure 4-19: Measured output velocity(solid) and the simulated output velocity(dashed) of the DC motor shaft in respond to a sinusoidal input. The simulation is made from the network model using the estimated physical parameters. The unit of the error shown is rad/sec.

The predicted response matches the phase of the actual response quite well. The error is largest at each velocity peak where the model over-estimates the speed by roughly 3 radi-

ans/second, which corresponds to a 10% prediction error. It can also be noticed that, at each measured peak velocity, a strange "chattering" was evident as the experiment progressed. This indicates that prior to reaching the maximum velocity, the motor decelerated a bit and accelerated up again. At the time of this writing, no satisfactory explanation of why the motor behaves in this way has been found. The network model is obviously incapable of producing this strange behavior and this results in the strange pattern of predicted error shown in the figure 4-19. Aside from some strange, non-linear behaviors of the motor (which cannot be reproduced by a linear model anyway), the overall predictions of the motor behaviors are acceptable. This comparison exercise and the results tabulated in the table 4.5 supports the use of the proposed network-based identification as a tool in mapping from observed behaviors to the geometric and material properties.

Chapter 5

Conclusions & Recommendations

In this thesis, the problem of constructing a map from behaviors to geometric and material parameters of a physical system via was tackled. Two approaches are used: "black-box" identification and physical structure-based identification. The "black-box" approach was found to be inadequate while the physical-structure approach looked promising.

5.1 Physical Structure is Necessary

From the results of the simulations in chapter 3 and the experiments in chapter 4, we have enough evidence to conclude that "black-box" identification is not adequate to synthesize a map from behaviors to geometric and material parameters of a physical system. With the "black-box" approach, we attempted to identify the motor parameters from the measurements of voltage and motor position. A DC motor was examined and considered as a relatively simple and linear-time-invariant system. However, the result was a failure. The motor parameters could not be uniquely identified. The mapping is inherently non-linear with the observed behaviors as non-linear functions of physical parameters. The approach provided no means to resolve the problems and left one to proceed in an ad-hoc way. Further, and even more disturbing, the derived "black-box" models were unstable at times.

With the proposed network-based identification (section 3.2), a different kind of result was obtained. The sought-for motor parameters were uniquely identified and the instability never occurred. The physical-structure of the motor (in the form of a network) provides necessary

information as to what motor behaviors should be observed to obtain adequate information about each physical parameter. The choice of behaviors that contains free integrators may be avoided with the network-based scheme.

Therefore, the knowledge of a physical structure is a necessary ingredient to the synthesis of a map from behaviors to physical parameters of a physical system. It is true that we considered only the case of a linear system and the algorithm developed reflects that fact. However, as illustrated, without physical structure, we could not even construct a sought-for map.

5.2 Possible Future Works

Our goal is to develop a systematic method of constructing a physically-parameterized model of a physical system from behavior. In the thesis, a network-based algorithm (section 3.2) is proposed which takes advantage of the network-structure (*port-structure*) of physical systems while also exploiting some techniques of the "black-box" approach. However, it has been applied to only a simple DC torque motor. It would be of great interest to test the algorithm on a different linear system with greater complexity; for example, consider an upper-arm amputation prosthesis emulator. It is an electro-mechanical system which contains the DC torque motor used in the experiment. Its network representation is shown in figure 5-1. It can be seen that the left portion of the emulator's network structure is in fact the structure of the DC motor, which we have repeatedly used. The emulator also consists of other physical devices such as a gear train, cable drive, and sensors. For complete details of the emulator and the model in figure 5-1, the reader is asked to refer to [1] and [2].

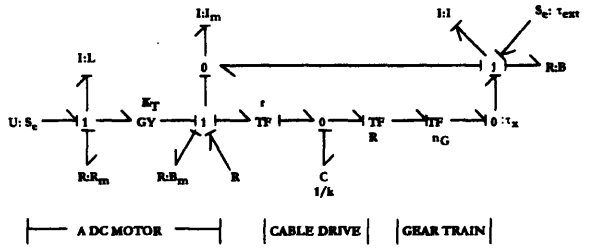

Figure 5-1: A network model of an upper-arm amputation prosthesis emulator

It would be interesting to find out whether the proposed algorithm can be used to determine the physical parameters that are included in the network model of the emulator. This exercise should serve as a sufficient test of the algorithm.

Early on in the thesis, It was mentioned that the network structure (or *port structure*) is modular and analogous to jigsaw pieces. Thus far, we have not used the modular property since the size of the network structure of the DC motor is small enough. To tackle the problem of determining the physical parameters of the emulator, taking advantage of the modularity of the network should be useful. How to exploit this modularity of a given network topology has not been fully worked out and is not given here. Certainly, it should be done in the near future. Its implementation, together with the proposed network scheme, should provide a user with (what could become) a comprehensive tool in mapping from behavior to geometric and material parameters of physical systems.

Appendix A

Validations of Model 3.5 and 3.6

The purpose of this appendix is to show the validation results of the model 3.5 and the model 3.6 discussed in chapter 3.

In the simulation, it turns out that results of the validation for both models are quite similar. Therefore, only results of the model 3.6 will be shown here. As a reminder, the model 3.6 and its corresponding continuous-time model is shown again here:

$$\dot{y} + \frac{f}{J}y = \frac{k}{J}i \tag{A.1}$$

$$y(t) + a_2 y(t-1) = b_2 i(t) \tag{A.2}$$

$$a_2 = -\frac{1}{1 + f/JT} \tag{A.3}$$

$$b_2 = \frac{kT/J}{1 + f/JT} \tag{A.4}$$

The behavioral parameters, a_2 and b_2 are obtained from the data set whose input is a white, Gaussian signal. To test the competency of the derived model, it is verified against two set of data: one is a unit step response and the other is a sinusoidal response. In both cases, the model performs quite well; the results of its performance are shown on figures A-1 and A-2

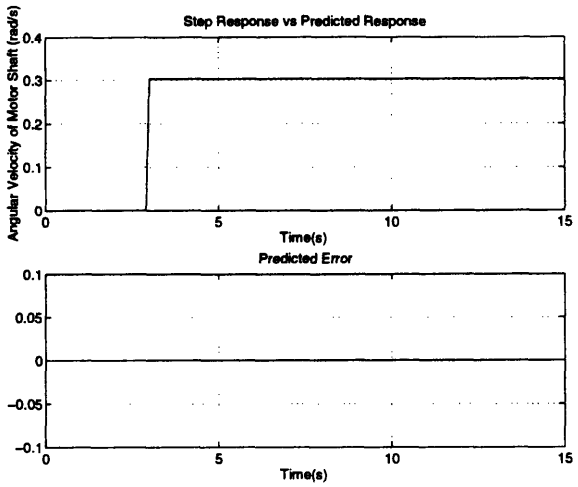

Figure A-1: The actual response vs. the predicted response of the model 3.6 to a step input. The y-axis of the upper plot is the angular velocity of the motor shaft (rad/sec). The unit of the error shown is rad/sec.

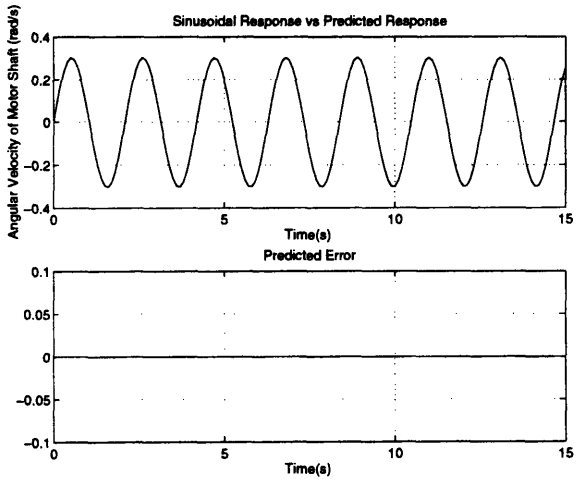

Figure A-2: The actual response vs. the predicted response of the model 3.6 to a sinusoidal input: $i = 5sin(6t)$. The y-axis of the upper plot is the angular velocity of the motor shaft (rad/sec). The unit of the error shown is rad/sec.

As compared with the random signal, the derived model also performs quite well; the result is shown in the figure A-3.

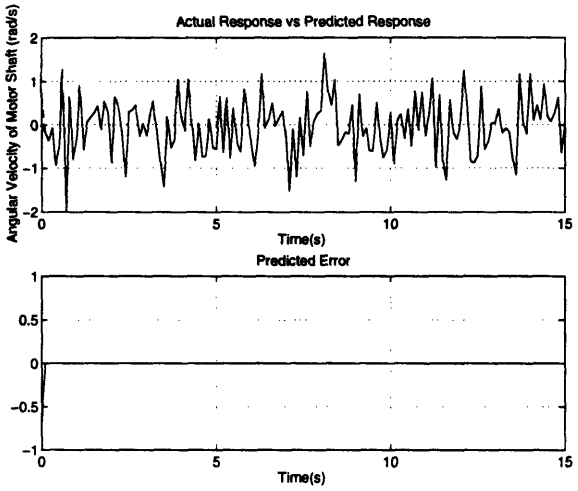

Figure A-3: The actual response vs the predicted response of the model 3.6 to a white Gaussian input. The unit of the error shown is rad/sec.

Appendix B

Data Simulation Using Matlab's *LSIM*

In section 3.1.1, the behavior of the DC motor was simulated by the *lsim* function in Matlab. This function simulates an ordinary differential equation by using either one of two algorithms referred to as *zero-order-hold* and *first-order-hold.* The former assumes the input data being held constant between sampling period while the latter linearly interpolates each input data point. The simulations in section 3.1.1 were done using the latter. Therefore, for the sake of completeness, the **A DC Motor Simulation 2** in section 3.1.1 is re-simulated using the *zero-order-hold algorithm (zoh).* The same voltage input data sequence is used again here. The motor output position is re-simulated and compared to that used in the **A DC Motor Simulation 2**. The results is shown in figure B-1.

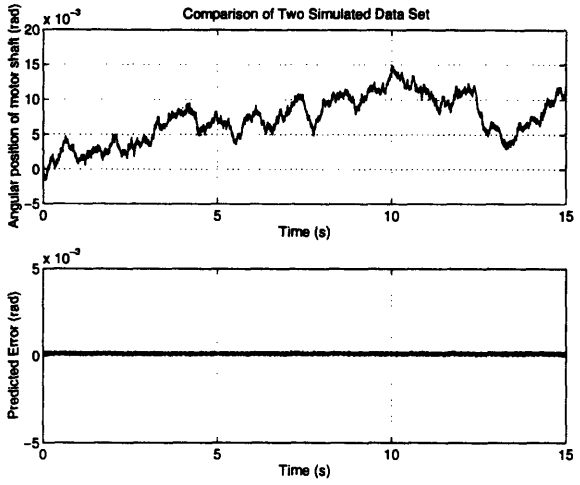

Figure B-1: Comparison between two simulated output where during one the input is linearly interpolated and during the other held constant between the sampling period.

As seen in the figure, there is not much difference between the two responses. The new data set is then used to construct the model of the form 3.1. The derived model is then used to reproduce the responses to a white Gaussian input and a step input; the results are shown below:

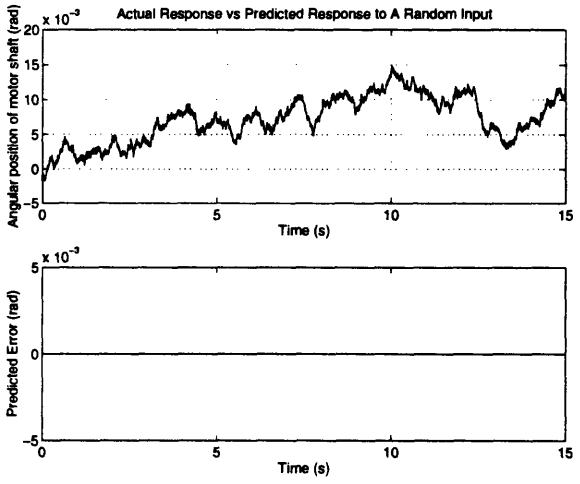

Figure B-2: Actual(simulated) response vs. predicted response of the model 3.1 to a white Gaussian signal. The simulation is done by the *zoh* algorithm.

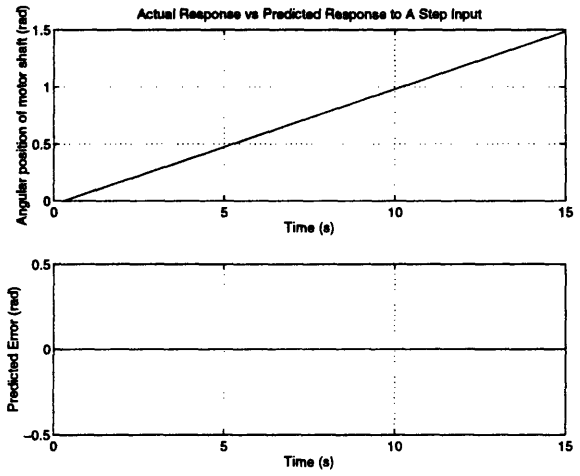

Figure B-3: Actual(simulated) response vs. predicted response of the model 3.1 to a step input. The simulation is done by the *zoh* algorithm.

It can be seen that the model can produce a better fit to the data than those shown in figures 3-5 and 3-6. An explanation to the improved reproduction of the simulated data is as follows. The set continuous-time models which is simulated here are the set of differential equations 2.11 and 2.12. Through the *ZOH transformation*, the corresponding discrete-time model is of the form identically to model 3.1. This is why such a model can reproduce the simulated data quite well here.

Through the *FOH*, the discrete-time model B.1 would fit better to the simulated data than those shown in **A DC Motor Simulation 2** in section 3.1.1.

$$y(t) + a_1 y(t-1) + a_2 y(t-2) = b_1 u(t) + b_2 u(t-1) \tag{B.1}$$

It should be noted that this additional information about *lsim* is interesting but it certainly doesn't change the results found in section 3.1.1.

Appendix C

Estimated Lower Bound of Motor Inertia

The purpose of this appendix is to show the assumptions and the procedure used in estimating the lower bound of the motor inertia. When put into operation, the rotor of the motor is attached to an aluminum cylindrical hub which houses a steel shaft. Both the hub and the shaft add more inertia into the motor. In addition, the encoder, which is used to measure the motor shaft position during the experiment, also adds some inertia. As a result, the estimated motor inertia from the experiment should be higher than that of the motor manufacturer.

To estimate a lower bound for the total motor inertia, the following assumptions are used:

- The total inertia of the motor mostly consists of that of the rotor, the encoder, the aluminum hub, and the steel shaft.
- The inertia of the rotor and the encoder is assumed to be that reported by its manufacturers.
- The geometry of the aluminum hub is treated as that of a thick ring to simplify the inertia calculation.

The dimensions of the hub and the steel shaft are shown in the table C.1.

Physical quantity	*Hub dimension (in.)*	*Steel shaft dimension (in.)*
Height/Length	0.17	2.50
Outer Diameter	1.35	0.25
Inner Diameter	0.25	N.A.

Table C.1: Dimension of the aluminum hub and the steel shaft

The mass of both hub and shaft is determined using the density value from [5]. To calculate the inertia, the geometry steel shaft is treated as a cylinder and the hub as a thick ring. The results of the estimated inertia of the motor and each component are tabulated in the table C.2.

Components	*Inertia* $(N - m - s^2)$
Rotor	$52.95e-06$*
Encoder	$0.71e-06$*
Steel Shaft	$1.46e-06$
Aluminum Hub	$1.56e-06$
Total	$56.68e-06$

Table C.2: Estimates of the inertia of the motor components. The * indicates the value taken from the manufacturer of the corresponding part.

The lower bound of the derived motor inertia is, of course, a rough estimate; the calculation of the inertia not only assumes simple geometry for certain components but also neglects the inertia due to the bearings on the motor shaft. However, it gives a new lower bound to which the inertia estimate from the identification procedure can be compared.

Bibliography

[1] Cary James Abul-Haj. The design of an upper-arm prosthesis simulator with variable mechanical impedance. Master thesis, Massachusetts Institute of Technology, Mechanical Engineering Department, September 1981.

[2] Cary James Abul-Jaj. *Elbow-Prosthesis Emulator: A Technique for The Quantitative Assessment of An Assistive Device.* PhD dissertation, Massachusetts Institute of Technology, Mechanical Engineering Department, June 1987.

[3] Gene F. Franklin, J. David Powell, and Michael L. Workman. *Digital Control of Dynamics Systems.* Addison-Wesley, Reading, Massachusetts, second edition, 1980.

[4] Neville Hogan and Peter Breedveld. Integrated modeling of physical system dynamics. Lecture Notes, July 17-21, 1995.

[5] Serope Kalpakjian. *Manufacturing Process for Engineering Materials*, section 3.8, page 130. Addison-Wesley, Reading, Massachusetts, second edition, April 1984.

[6] Lennart Ljung. *System Identification: Theory for the User.* Prentice Hall information and system sciences. Prentice Hall, Englewood Cliffs, New Jersey, 1987.

[7] Lennart Ljung and Torkel Glad. *Modeling of Dynamics Systems.* Prentice Hall information and system sciences. Prentice Hall, Englewood Cliffs, New Jersey, 1994.

[8] Ahmed H. Mitwalli. Parameter estimation using least-squares and maximum likelihood criterion- area exam report. Technical report, Massachusetts Institute of Technology, Electrical Engineering and Computer Science Department, Cambridge, Massachusetts, September 1995.

[9] N. K. Sinha. Estimation of transfer function of continuous system from sampled data. *Im EE Proc. Journal*, 119:612–614, 1972.

[10] In N.K. Sinha and G.P. Rao, editors, *Identification of Continuous-Time Systems: Methodology and Computer Implementation.* Kluwer Academic Publishers, 1991.

[11] P. C. Young. Parameter estimation for continuous-time models. *Automatica Journal*, 17:23–29, 1981.